DOCUMENT RESUME

ED 088 920 TM 003 476

AUTHOR Durward, M. Lynn; Ellis, E. N.
TITLE An Evaluation of the Computer Science 11E Course Offered During the Spring Semester of 1973.
INSTITUTION Vancouver Board of School Trustees (British Columbia). Dept. of Planning and Evaluation.
REPORT NO RR-73-21
PUB DATE Jul 73
NOTE 92p.

EDRS PRICE MF-$0.75 HC-$4.20
DESCRIPTORS *Computer Science Education; *Course Evaluation; Questionnaires; *Secondary School Students; Student Attitudes; Student Evaluation; Teacher Evaluation
IDENTIFIERS Canada

ABSTRACT

The evaluation of Computer Science 11E, a locally-developed course offered in six Vancouver secondary schools, consisted of four parts: a questionnaire to teachers concerning all aspects of their courses, as well as their own evaluation of the program, a questionnaire to students designed to determine their attitude toward the course and what they considered to be the strengths and weaknesses of the program, a set of programming problems developed to obtain objective evidence of the students' problem-solving ability, and a report on the impressions gained by an independent observer during a visit to one of the computer science classes. The results showed that the objectives of the Computer Science 11E course had been reached. The performance of the students on the programming problems indicated that their grasp of both the BASIC (Beginner's All-Purpose Symbolic Instruction Code) programming language and problem-solving techniques were good. Students willingly applied the knowledge gained in the computer classes to other areas; more than one-half of them wrote programs to solve problems in other courses. Both teachers and students suggested that more hands-on computer time be made available, and that other programming languages, in addition to BASIC be included in the course content. For related documents, see TM003469 and TM003475. (Author/RC)

ED 088920

TM 003 476

RESEARCH REPORT

An Evaluation of the Computer Science 11E Course Offered During the Spring Semester of 1973

July, 1973

M. Lynne Durward

and

E. N. Ellis

Research Report 73-21

DEPARTMENT OF PLANNING AND EVALUATION
Board of School Trustees
1595 West 10th Avenue
Vancouver 9, B.C.

ED 088920

AN EVALUATION OF THE COMPUTER SCIENCE 11E COURSE OFFERED DURING THE SPRING SEMESTER OF 1973

July, 1973

M. Lynne Durward
and
E. N. Ellis

Research Report 73-21

TABLE OF CONTENTS

	Page
ABSTRACT	i
INTRODUCTION	1
OUTLINE OF THE STUDY	2
ANALYSIS OF RESPONSES TO THE TEACHER QUESTIONNAIRE	3
ANALYSIS OF RESPONSES TO THE STUDENT QUESTIONNAIRE REGARDING COMPUTER-BASED INSTRUCTION	12
THE PROBLEM-SOLVING ABILITY OF COMPUTER SCIENCE 11E STUDENTS	16
OBSERVATION OF A COMPUTER SCIENCE 11E CLASS	18
SUMMARY	18
BIBLIOGRAPHY	20

APPENDIX A: References and Guide for the Computers 11 Course
APPENDIX B: Questionnaire for Teachers About Computer-Based Instruction
APPENDIX C: Student Questionnaire Regarding Computer-Based Instruction
APPENDIX D: Set of Programming Problems
APPENDIX E: The Computer Programmer Aptitude Battery
APPENDIX F: A Previous Study of Computer-Based Instruction in Vancouver Secondary Schools
APPENDIX G: Survey of Student Usage of the Hewlett-Packard Computers
APPENDIX H: Tic Tac Toe Game
APPENDIX I: Lunar Module Game
APPENDIX J: Catalogue of References and Aids for Hewlett-Packard Computer Users
APPENDIX K: Examples of Programs Written by Computer Science 11E Students

LIST OF TABLES AND FIGURES

Tables		Page
I	SUMMARY OF QUALIFICATIONS OF COMPUTER SCIENCE TEACHERS	7
II	SUMMARY OF THE COMPUTER EXPERIENCE OF STUDENTS ENROLLED IN COMPUTER SCIENCE 11E CLASSES	7
III	WEIGHTED-MEAN PRIORITIES AND NUMERICAL RANKS ASSIGNED TO SUBJECT AREAS BY TEACHERS TO INDICATE AREAS OF PROBLEMS ON WHICH THE STUDENTS WERE WORKING	9
IV	SUMMARY OF CLASS ACTIVITIES PROVIDED FOR COMPUTER SCIENCE 11E CLASSES	10
V	SUMMARY OF RESPONSES TO THE STUDENT QUESTIONNAIRE REGARDING COMPUTER-BASED INSTRUCTION (N=71)	12
VI	SUMMARY OF NUMBER OF COMPUTER RUNS REQUIRED BY COMPUTER SCIENCE STUDENTS (N=71) TO SOLVE A PROGRAMMING PROBLEM	17
VII	SUMMARY OF TYPES OF ERRORS MADE BY COMPUTER SCIENCE STUDENTS WHILE TRYING TO SOLVE A PROGRAMMING PROBLEM	17

FIGURE 1: Percentage of Class Time Spent by Teachers with Individuals, Small Groups, and Entire Class 8

AN EVALUATION OF THE COMPUTER SCIENCE 11E COURSE OFFERED DURING THE SPRING SEMESTER OF 1973

Abstract

The evaluation of Computer Science 11E, a locally-developed course offered in six Vancouver secondary schools, consisted of four parts:

1. a questionnaire to teachers concerning all aspects of their courses, as well as their own evaluation of the program,

2. a questionnaire to students designed to determine their attitude toward the course and what they considered to be the strengths and weaknesses of the program,

3. a set of programming problems developed to obtain objective evidence of the students' problem-solving ability, and

4. a report on the impressions gained by an independent observer during a visit to one of the computer science classes.

The results of the study showed that the objectives of the Computer Science 11E course, as stated by the teachers, had been reached. Students felt they had gained a sound knowledge of computers in general, and an understanding of their sociological impact. The performance of the students on the programming problems indicated that their grasp of both the BASIC programming language and problem-solving techniques were good.

The teachers of Computer Science 11E were generally highly qualified, in terms of level of education and previous experience with computers, to teach the subject matter.

Teachers described their students as being self-motivated and of above-average scholastic aptitude. The findings of the student questionnaire and the impressions gained during a visit to one of the classes supported this contention.

The Computer Science 11E classes appeared to be conducted on a more informal basis than were regular classes. Students were encouraged to work independently and at their own speed. Teachers spent almost 60% of their class time working with individuals.

Students willingly applied the knowledge gained in the computer classes to other areas; more than one-half of them wrote programs to solve problems in other courses

Both teachers and students suggested that more hands-on computer time be made available, and that other programming languages, in addition to BASIC, be included in the course content.

AN EVALUATION OF THE COMPUTER SCIENCE 11E COURSE OFFERED DURING THE SPRING SEMESTER OF 1973.

INTRODUCTION

Few Canadians are aware of the pervasive influence of computers. Computers are used to control the timing of traffic lights in Toronto, to dispatch taxicabs to riders in Montreal, to check income tax returns in Ottawa and to confirm airplane reservations across Canada.

The influence of the computer is not restricted to the adult population. Here in Vancouver, for instance, secondary school students are now receiving computer-produced individualized timetables and report cards, and selected elementary school children are receiving computer-assisted instruction.

With the use of computers becoming so widespread, educators throughout the world have become aware of the need for computer-based instruction. In the United States, the Conference Board of Mathematical Science's Committee on Computer Education concluded:

> It is therefore essential that our educational system be modified in such a way that every student (i. e., every prospective citizen) become acquainted with the nature of computers and the current and potential roles which they play in our society. It is probably too late to do much about adults, but it would be disastrous to neglect the next generation. [1]

The next generation is not being neglected in Vancouver secondary schools. During the spring semester of 1973, Computer Science 11E, a locally developed course approved by the Department of Education in Victoria, was offered in six schools: Lord Byng, Templeton, Tupper, Kitsilano, Point Grey and John Oliver secondary schools.

The Computer Science 11E course was developed by Mr. Wayne Dodds, Computer Consultant for the Vancouver School Board. In his References and Guide for the Computers 11 Course, (see Appendix A, p. 1), Mr. Dodds described what he considered to be the objectives of the course: to provide the students with a "conceptual understanding, a little fundamental knowledge, but very limited skills, regarding the computer's

1) application areas of social significance,

2) potential and limitations, and

3) fundamentals of hardware and software."

---

[1] Committee on Computer Education, Conference Board of the Mathematical Sciences, "Recommendations Regarding Computers in High School Education", p. 3.

The guide also includes a detailed outline of the topics set out to be covered in the course.

In December of 1972, a committee was formed to propose an evaluation of Computer Science 11E. The committee was comprised of two members of the Department of Planning and Evaluation of the Vancouver School Board, Mr. Wayne Dodds, and three computer science teachers. This committee drafted what it considered to be the objectives of computer-based instruction:

1) To enable students to gain some appreciation of computers.

2) To develop the ability of students to reason logically, and in sequence, from cause to effect.

3) To develop in students the ability to read directions and to follow instructions.

4) To develop in students the ability to apply knowledge to new situations.

5) To cause students to work systematically and to be orderly in their work habits.

6) To increase the efficiency of student performance.

7) To provide students with opportunities to experience success and satisfaction in getting the computer to do what they want it to.

8) To develop a better student-teacher relationship.

9) To develop in students the ability to work well with others.

10) To improve the student's self-concept by causing him to be self-reliant and to have more confidence in his own ability because of the facility of the computer to verify his solutions.

Using these objectives as a guide, the committee developed several instruments to evaluate Computer Science 11E (see next section). This report summarizes the findings of the study subsequently carried out.

OUTLINE OF THE STUDY

The evaluation of Computer Science 11E consisted of four parts:

1) A questionnaire (see Appendix B) was sent to teachers of Computer Science 11E in six secondary schools to collect information about teacher background and experience, student computer experience, course content, method of instruction, use of computer facilities, use of reference materials, enrichment activities, and teacher evaluation of the program.

2) The attitudes of the students enrolled in the computer science classes

toward the course were examined by means of a questionnaire (see Appendix C). The students were encouraged to state what they saw as the strengths and weaknesses of the program and to offer suggestions for its improvement.

3) Students were asked to complete a programming problem (see Appendix D) in order to obtain objective evidence of their problem-solving ability. A record was kept of the type of errors made by each student and the number of computer runs each required to solve the problem.

4) An independent observer was asked to visit one of the computer science classes to observe the interaction among the students and teacher, to take note of the activities of the students, and to interview a few students to verify and extend the impressions obtained from the analysis of responses to the questionnaires.

Five of the six schools offering Computer Science 11E courses participated in the study. The instructor at John Oliver Secondary School was unable to take part because of prolonged illness.

Three additional items of interest, not directly concerned with Computer Science 11E courses, but related to computer courses in general, are included in the appendices. Appendix E contains a report on the performance of a class of computer science students on the Computer Programmer Aptitude Battery. Appendix F contains the results of a recent survey of student usage of the computer facilities. Appendix G contains a summary of a previous study on computer courses in Vancouver Secondary schools.

ANALYSIS OF RESPONSES TO THE TEACHER QUESTIONNAIRE

Teachers of Computer Science 11E in five schools completed the questionnaire (see Appendix B). A total of 77 students were enrolled in the five classes included in the study.

Ultimate goal of the course

The ultimate goal of the course, in the opinion of the teachers, was to:

a) present an orientation of the use of computers in business and society,
b) develop an awareness of the sociological impact of information processing,
c) introduce a programming language, and
d) introduce students to problem-solving techniques using the computer.

Topics and objectives

Although the topics covered varied slightly among schools, four main ones were cited by the teachers:

Topics	Objectives
a) Introduction to computers	- to teach students what a computer is, how it works, and what its applications are
b) Flowcharting and problem-solving techniques	-to show students how to use flowcharts to analyze and solve problems
c) Programming in BASIC language	-to teach students how to use BASIC language to solve problems by computer
d) Social implications of computers	-to show how some areas of society may benefit or suffer from the use of the computer.

Use of computer facilities

The Vancouver School Board owns two Hewlett-Packard digital computers which are used by the students in the Computer Science 11E courses. The main input device to each computer is a mark sense card reader. Students write their programs in BASIC language on the "mark sense" cards and submit them for "batch" processing. (A "batch" of computer programs refers to two or more students' programs which are combined into one deck of cards and run through the computer consecutively by the computer operator). A pickup and delivery service is provided to transmit the batches of programs from surrounding schools to the computer facilities at either John Oliver or Point Grey Secondary Schools. The results from the programs (output) are printed by a high speed line printer and are usually returned to students in time for their next class session in that subject.

The pickup/delivery service was used by all the students in the computer science classes it served, and most students used it on a daily basis.

The disadvantage in using the pickup/delivery service for batch mode BASIC was that students had to wait a day to get their programs back and correct errors. Thus, if a student had five errors in his program, it might take him a week before he had discovered and made all the necessary corrections. By using "hands-on batch mode" BASIC in the evenings, students could overcome this problem. Evening time had been reserved on the two Hewlett-Packard computers so that students could come to the installation, submit their programs directly to the computer and receive their output almost instantaneously. Programs usually were submitted on optic mark sense cards but occasionally were submitted on punched paper tape (which was read by a high speed photo-electric paper tape reader). The output from the computer was printed onto a sheet of paper, or, in special circumstances, onto paper tape.

According to the teachers, 71.4% of all the students in the computer science classes used the hands-on batch mode in the evenings with mark sense cards;

15.6% used paper tape programs. The frequency of evening use varied among the schools. Most classes had weekly evening sessions; however, the computer science 11E students from the school with the computer used the facilities on almost every available evening of the month.

Students could communicate directly with the computer via a portable terminal which was used by the schools on a rotating basis. The advantage to this "hands-on interactive coding" was that a student could correct his errors while his program was in the memory of the computer. No cards or paper tapes were necessary, - programs were typed in on the terminal keyboard. Two of the five computer science classes used the terminal, one class on two occasions during the term, the other on ten occasions.

Three of the classes used the portable terminal with demonstration and other centrally-supplied BASIC programs. These "packaged" programs enabled students to try to outwit the computer in a number of educational or motivational games. Two such games, the "Tic Tac Toe Game" and the "Lunar Module Game" are shown in Appendices H and I.

Six students programmed in APL: (A Programming Language) for a period of six weeks. The portable terminal they used was connected (via telephone lines) to the computing facilities at Simon Fraser University.

Reference materials used by students

Teachers were requested to list the reference materials (catalogue items, books and other materials) used by their students and to indicate the frequency of their use.

The "catalogue items" consisted of reference sheets on various aspects of the BASIC language and general use of the computer, and assorted teacher aids (e.g. coding sheets, lists of movies on data processing, etc.). They were supplied by Mr. Dodds and were available through Point Grey and John Oliver computing centres. The original list of catalogue items was compiled with the aid of computer science teachers. Of the 29 catalogue items available (see Appendix J), those used most frequently by students were:

Catalogue Item	No. of Students Who Used It (N=77)
"How to Use Alpha Data"	65
"Disk Files for Educational BASIC"	62
"Error Codes... Educational BASIC"	53
"A Guide to HP Educational BASIC"	37
"Disk"	32

Teachers cited a total of six books that were used by their students for reference. Of these, two were used frequently in most classes: "A Pocket Guide to Hewlett-Packard Computers" produced by the Hewlett-Packard Company, and the "Tecnica Series" of books produced by the Tecnica Education Corporation.

Among the other materials that teachers specified as being used by students were CARDIAC (Cardboard Illustrative Aid to Computation - a small do-it-yourself cardboard computer), worksheets prepared by the teachers, the teachers' own outlines and notes, and programming problems prepared by Mr. Dodds for the Computers 11 course.

Reference materials used by teachers

The catalogue items used for reference most frequently by teachers are listed below.

Catalogue Item	No. of Teachers Using It (N=5)
"Error Codes ... Educational BASIC"	3
"How to Use Alpha Data"	3
"Disk Files for Educational BASIC"	3
"Disk"	2

The frequency of use varied considerably among teachers.

Seven books which were used for reference were cited by the teachers. No two teachers used the same book.

Percent of teaching time spent instructing computer courses

The percent of teaching time (with respect to all regular courses as well as computer courses) spent instructing Computer courses ranged from 14.3% to 50%; the average percent of time spent was 29.3%.

Qualifications of teachers of Computer Science 11E courses

A summary of the qualifications of Computer Science 11E teachers is presented in Table I. Two of the five teachers held Masters Degrees in addition to Bachelor Degrees; the median number of computer science courses taken by the teachers was three; and most of the teachers had considerable background experience with computers.

TABLE I: SUMMARY OF QUALIFICATIONS OF COMPUTER SCIENCE TEACHERS.

Teacher	Teaching Certificate	University Degrees	Computer Science Courses Taken	Background Computer Experience
1	P. B.	B. Ed.	3	Taught previous Computer courses at high school, college and university level; worked as computer operator.
2	P. B.	B. Sc.	4	None (excluding courses)
3	P. A.	B. A., B. Sc.	3	Worked as computer programmer.
4	P. A. Masters	B. Ed., M. Ed.	1	Worked as coordinator of student scheduling; taught previous computer science courses at high school level.
5	P. A. Masters	B. Ed., M. Ed.	2	Used the computer for instructing mathematics; considerable experience in graduate courses in computer techniques in the administrative process.

Computer experience of students enrolled in Computer Science 11E

Students from grades 9 to 12 were enrolled in Computer Science 11E classes. The largest percentage of these (37.7%) were from Grade 11. Most of the students (87.0%) had had no previous experience with computers; a few had taken computer courses previously (see Table II).

TABLE II: SUMMARY OF THE COMPUTER EXPERIENCE OF STUDENTS ENROLLED IN COMPUTER SCIENCE 11E CLASSES.

Grade Level	Number of years of computer experience				
	0	1	2	Total	%
9	15	-	-	15	19.5
10	6	6	-	12	15.6
11	25	-	4	29	37.7
12	21	-	-	21	27.3
Total	67	6	4	77	100.0
%	87.0	7.8	5.2		

General characteristics that applied to students enrolled in Computer Science 11E

In general, students in Computer Science 11E classes were described by teachers as being self-motivated and of above-average scholastic aptitude.

Amount of teaching time spent with individuals, small groups, and the entire class

Computer Science 11E teachers, on the average, spent the largest part of their class time (57%) working with individuals (see Figure 1).

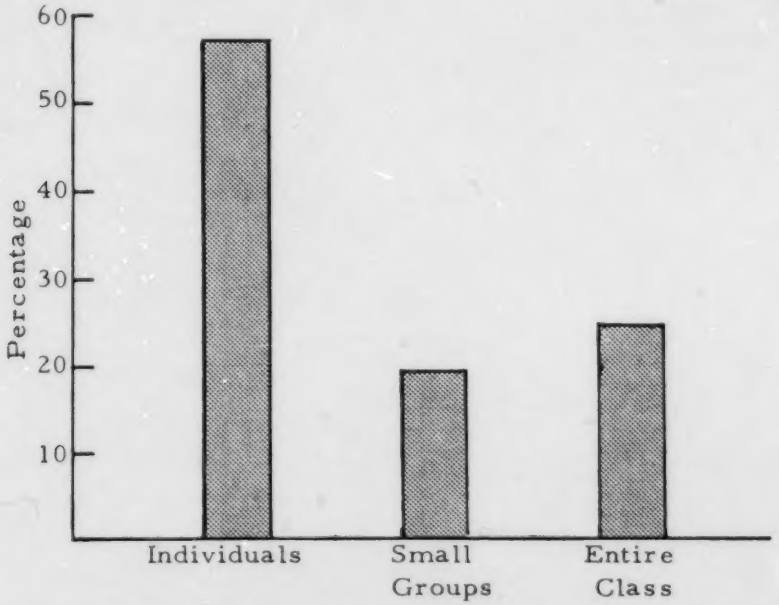


FIGURE 1: PERCENTAGE OF CLASS TIME SPENT BY TEACHERS WITH INDIVIDUALS, SMALL GROUPS, AND ENTIRE CLASS.

Procedures used to evaluate student achievement

Teachers were asked to describe the procedures they used to evaluate student achievement. Their responses are listed below:

Teacher 1: "Working programs, effort and interest, objective theory tests, occasional subjective tests, projects, classwork."

Teacher 2: "Stage of progress, willingness to stick to it until they themselves get a solution, method of solution, initiative."

Teacher 3: "Work habits only (no grades given)."

Teacher 4: "Assignments - complete two programs from five or six possibilities,
plus a written examination, testing: knowledge of commands, recognition of errors in sample programs, and writing a program under a time limit, and
overall progress and knowledge displayed in class work."

Teacher 5: "Degree of accurate completion of set problems,
Degree of initiative in application of techniques to topics of the student's own interest, and
Degree of involvement in non-programming objectives of the course."

Subject areas in which computer science students were working on problems

Teachers assigned numerical ranks to subject areas to indicate those areas in which their students were working most frequently. From these ranks, weighted mean priorities were calculated, and the priorities were then assigned numerical ranks (see Table III). Students were found to be working on problems in many of their school subjects; mathematics received the highest ranking.

TABLE III: WEIGHTED-MEAN PRIORITIES AND NUMERICAL RANKS ASSIGNED TO SUBJECT AREAS BY TEACHERS TO INDICATE AREAS OF PROBLEMS ON WHICH THE STUDENTS WERE WORKING.

Subject Area	Weighted Mean Priority	Numerical Rank
Mathematics	1.6	1
Physics	2.6	2
Commerce	3.2	3
Science	3.5	4
Biology	3.5	4
Chemistry	4.0	6
Games	4.2	7

Class activities provided

A summary of the class activities provided for computer science classes is presented in Table IV. Audio-visual presentations were the most frequently provided class activities. They were presented, on the average, thirteen times per term, and involved 61.0% of the students enrolled in the computer science classes. Discussions of newspaper clippings were held less frequently (10.6 times per term) but involved more students (72.7%).

TABLE IV: SUMMARY OF CLASS ACTIVITIES PROVIDED FOR COMPUTER SCIENCE 11E CLASSES.

Activity	Average Number of Times Provided Per Class	Percentage of Students Participating
Audio-Visual Presentation	13.0	61.0
Newspaper Clippings	10.6	72.7
Magazine Articles	8.6	29.9
Research Assignments	1.4	32.5
Field-trips	0.4	7.8

Strengths of the computer-based instructional program

Teachers cited the following as strengths of the Computer Science 11E program:

> "Because it's an option, only students who wanted to take the course took it."
>
> "Freedom of approach."
>
> "Good achievement-oriented outlet for bright students; an alternative for non-academic students in mathematics and commerce."
>
> "Opportunities for 'hands-on' time and a chance to learn actual start-up and operational routines of the normal running of the computer."
>
> "Has provided mathematics/science oriented students with an elective of import related to their area of interest."
>
> "Has forced students taking the course to centre their concern on problem-solving techniques applicable to many subject areas."

Weaknesses of the computer-based instructional program

The following weaknesses of the program were cited by teachers:

> "Have not as yet moved out of the school enough, i.e. lack of field trips, 'hands-on' time on computer, etc."

"Have not developed a wide enough range of materials to meet needs of students entering the course with wide diversity in educational background."

"Only good for bright academic students."

"Students should be taught some programming before Grades 11 and 12."

"There are not enough supplementary exercises available."

"No other languages (besides BASIC) are taught."

"Students have little direct access to hardware."

"The computer cannot accommodate FORTRAN or languages other than BASIC. These are of particular value for advanced and interested students."

"Having all four grades (9 to 12) in a beginning class creates some problems -- a great gap in ability to reason out problems and ability to concentrate on a problem for a lengthy period of time."

Suggestions for improvement and extension of the course

Teachers made the following suggestions for improvement and extension of the course:

"Additional course for non-academic commercial students."

"More reference material, higher level of supplementary exercises."

"More hardware and software support."

"More integration with various departments whose students could benefit from the present computer facilities -- this is in fact now beginning to happen in Science, Mathematics, Commerce and Social Studies."

Examples of problems on which the computer science students were working

Teachers were asked to include sample printouts of an easy, an average, and a difficult problem attempted by students in the computer science class they taught. Three such examples are listed below:

1. "Address labels" (easy) -- This program prints labels which include a name, address and a telephone number.

2. "Pay cheques" (average) -- This program prints out company pay cheques and checks for invalid employee numbers.

3. "Volume of a sphere" (difficult) -- This program calculates the volume of a sphere by dividing the sphere into rectangular solids (simulates the 'calculus' method).

Appendix K contains the printed output from each of the above examples.

ANALYSIS OF RESPONSES TO THE STUDENT QUESTIONNAIRE REGARDING COMPUTER-BASED INSTRUCTION

Questionnaires were distributed to the 77 students enrolled in Computer Science 11E classes; they were returned by 71 (92.2%) students.

Table V summarizes the students' responses to the questionnaire regarding computer-based instruction.

TABLE V: SUMMARY OF RESPONSES TO THE STUDENT QUESTIONNAIRE REGARDING COMPUTER-BASED INSTRUCTION (N=71)

1. In respect to difficulty, how did you find the computer course?
   - 23.9% (A) easier than I had expected
   - 9.9% (B) harder than I had expected
   - 66.2% (C) about the same as I had expected

2. The work load involved in the computer course was:
   - 5.6% (A) heavier than I had anticipated.
   - 39.4% (B) lighter than I had anticipated.
   - 54.9% (C) about the same as I had anticipated.

3. The course itself was:
   - 31.0% (A) highly interesting.
   - 62.0% (B) fairly interesting.
   - 7.0% (C) not interesting.

4. How many hours per week did you spend (outside of class time)
   - 1.0 hours (median) (A) on required work for the computer course?
   - 1.1 hours (median) (B) on computer work for your own interest and/or enjoyment?

5. On the average, did you spend more or less time outside of class on the computer course than on other courses?
   - 14.1% (A) more
   - 67.6% (B) less
   - 16.9% (C) about the same
   - 1.4% (D) no response

6. If you spent more time, what accounted for the extra time?

\- (A) heavy work load

70.0% (B) interest in the course

20.0% (C) other (Specify) _____

10.0% -- No response

7. Have you had sufficient opportunities to get help from your teacher? (Check one and comment if necessary)

85.9% (A) Yes

12.7% (B) No Comment _____

1.4% -- No Response

8. Rank the following according to their importance as sources of help in the computer course. (i.e., Insert "1" for the source that helped you the most, "2" for the second most important source of help, etc.) Any additional sources of help you consider important but are not listed, may be entered beside "Other" and ranked accordingly.

2 other students

1 teachers

4 reference material

5 other _____

3 computer error messages

9. Did you write programs for other courses?

(A) as assignments from teachers? (Check one)

54.9% Yes Name the courses Science (36.6%) Mathematics (28.2%)

43.7% No

1.4% -- No response

(B) on your own initiative? (Check one)

59.2% Yes Name the courses Science (31.0%) Mathematics (28.2%)

35.2% No

5.6% -- No response

Comment, if necessary_____

10. Have the thinking processes that you have developed in the computer course helped you in any other courses?

38.0% Yes

47.9% No

14.1% -- No response

What courses? Science (23.9%) Mathematics (23.9%)

How? Helped me to think logically (13.3%)

11. Please list what you consider to be the strengths and/or weaknesses of the computer course. In addition list any suggestions you might have to improve the course.

Strengths

1. "The course provided me with a general knowledge of computers." (38.0%)
2. "Good teacher and good instruction." (14.1%)
3. "It helped develop logical thinking." (11.3%)
4. "It helped in other courses." (11.3%)
5. "The course is interesting." (9.9%)

Weaknesses

1. "Not enough 'hands-on' computer time." (15.5%)
2. "The course is boring." (11.3%)
3. "The only programming language taught is BASIC." (9.9%)
4. "The turn-around time is too long." (8.5%)
5. "Card-marking is tedious." (7.0%)

Suggestions for improvement

1. "Better access to and more time on computer." (21.1%)
2. "Teach other programming languages." (7.0%)
3. "Get our own computer." (3.6%)
4. "More choice in selecting problems." (4.2%)
5. "Better computers." (2.8%)

12. The use of computers is increasing.

(a) How is this good for people?

"Computers are quicker and more efficient." (46.5%)

"Computers provide people with more leisure time." (7.0%)

(b) How is this bad for people?

"Computers put people out of work." (35.2%)

"People become lazy and too dependent on computers." (19.7%)

13. How will this course help you in the future?

"It might lead to a job in the computer field." (21.1%)

"It gives an insight into computers and programming." (14.1%)

"It won't" (12.7%)

"It will help in university or higher level courses" (4.2%)

14. Do you plan to pursue a career in the computer field?

14.1%(A) Yes

29.6%(B) No

53.5%(C) Undecided

2.8% -- No response

15. On your last report card, what was your letter grade standing in the computer course? "B" (median)

Students in general spent less time on homework for their computer science course than for other courses. Of those who indicated that they spent more time on the computer course, most cited interest in the course as the reason; none attributed any additional time spent to a heavy work load.

The programming skill acquired in computer science 11E had been applied in other areas--the majority of students wrote programs as assignments from teachers (most often in the areas of mathematics and science), and over half of the students wrote programs for other courses on their own initiative.

Teachers were ranked as the source of the greatest amount of help in the computer course. This finding was consistent with the finding that teachers spent most of their class time working with individuals.

Students cited the general knowledge of computers gained in the course as the main strength of the program. They suggested that better access to, and more time on the computer be provided.

Computer science students felt that computers were good in that they were quicker and more efficient at performing tasks than their human counterparts, but they were concerned about the possibility of computers putting people out of work.

Although the majority of students were undecided as to whether they would pursue a career in the computer field, 14.1% already had decided to enter the field.

The median letter grade ("B") received by students in the computer science course supported the contention of the teachers that the students enrolled in the classes were of above-average academic ability.

THE PROBLEM-SOLVING ABILITY OF COMPUTER SCIENCE 11E STUDENTS

A committee comprised of computer science teachers developed a set of problems (see Appendix D) in programming in order to obtain objective evidence of the problem-solving ability of students. The problem sets were given out after the students had completed a minimum of 60 hours of instruction. Each student was asked to select a problem (in one of the areas of mathematics, commerce or science), write a computer program to solve it, and get the program to run (without errors) on the computer. Teachers kept track of the number of computer trial runs the student required to complete his assignment and the type of errors each committed.

A summary of the number of trial runs required by the students to complete the assignment is presented in Table VI. The majority of students (67.6%) solved the problem in three or fewer runs; this was considered to be evidence of good problem-solving ability. Most of the teachers allowed a maximum of five runs; 14.1% of the students did not complete the assignment within this limit.

A breakdown of the type of errors committed by the students is presented in Table VII. Over half of the errors made were errors in the logic of the students' programs; 23.2% were "syntax" errors (in punctuation, spelling, etc.). Card-marking errors accounted for 17.1% of the total, and 3.1% of the errors were attributed to machine errors (e.g., the misreading of a card by the mark sense card reader).

TABLE VI: SUMMARY OF NUMBER OF COMPUTER RUNS REQUIRED BY COMPUTER SCIENCE STUDENTS (N=71) TO SOLVE A PROGRAMMING PROBLEM

Number of Runs Required	Percentage of Students (N = 71)
1	12.7%
2	23.9%
3	31.0%
4	12.7%
5	5.6%
Unsuccessful in 5 runs	14.1%
Total	100.0%

TABLE VII: SUMMARY OF TYPES OF ERRORS MADE BY COMPUTER SCIENCE STUDENTS WHILE TRYING TO SOLVE A PROGRAMMING PROBLEM

Type of Error	Percentage of Total No. of Errors
logic	53.9%
syntax	23.2%
card marking	17.1%
machine	3.1%
miscellaneous	2.6%
Total	100.0%

OBSERVATION OF A COMPUTER SCIENCE 11E CLASS

An independent observer visited one of the Computer Science 11E classes in order to take note of the activities of the students, to observe the student-teacher interaction, and to interview both teacher and students to extend and verify the impressions obtained from the analyses of responses to the questionnaires.

On the day of the visit, no formal lecture was given; the main reason being that the portable terminal was in the school for the day and the teacher preferred the class to devote its class time to using it. A few students left the classroom to operate the terminal: the remainder of the group quickly settled down to individual work. The teacher spent the balance of the class period assisting students with their problems. A high level of teacher-student rapport was evident, as was enthusiasm on the part of the students, especially among those operating the terminal.

Asked about the general characteristics of the students, the teacher commented that they were all of "A" or "B" calibre and were "very keen, bright and independent". Each had been required to work on his own computer problem "from scratch". One student was teaching the computer to play bridge, another was developing a personality questionnaire to be interpreted by computer.

The students, according to the teacher, were extremely enthusiastic about having 'hands-on' time at John Oliver Secondary School. During the first week there an operator had shown them how to run the computer; they had been operating it themselves since then.

It was the teacher's opinion that there should be a strict division of courses--one for data processing and one simply for programming: "Keen students get bored with data processing and slower students get lost with programming". She felt that a more business-oriented language (e. g. FORTRAN or COBOL) should be taught in addition to BASIC. Further remarks confirmed those she reported on the teacher questionnaire.

Students questioned had one main criticism of the Computer Science 11E program: "The computer is not large enough to handle our problems--it runs out of memory when running our programs."

SUMMARY

The evaluation of Computer Science 11E, a locally-developed course offered in six Vancouver secondary schools, consisted of four parts:

1) a questionnaire to teachers concerning all aspects of their courses, as well as their own evaluation of the program,
2) a questionnaire to students designed to determine their attitude toward the course and what they considered to be the strengths and weaknesses of the program,
3) a set of programming problems developed to obtain objective evidence of the students' problem-solving ability, and

4) a report on the impressions gained by an independent observer during a visit to one of the computer science classes.

The results of the study showed that the objectives of the Computer Science 11E course, as stated by the teachers, had been reached. Students felt they had gained a sound knowledge of computers in general, and an understanding of their sociological impact. The performance of the students on the programming problems indicated that their grasp of both the BASIC programming language and problem-solving techniques were good.

The teachers of Computer Science 11E were generally highly qualified, in terms of level of education and previous experience with computers, to teach the subject matter.

Teachers described their students as being self-motivated and of above-average scholastic aptitude. The findings of the student questionnaire and the impressions gained during a visit to one of the classes supported this contention.

The Computer Science 11E classes appeared to be conducted on a more informal basis than were regular classes. Students were encouraged to work independently and at their own speed. Teachers spent almost 60% of their class time working with individuals.

Students willingly applied the knowledge gained in the computer classes to other areas; more than one-half of them wrote programs to solve problems in other courses.

Both teachers and students suggested that more 'hands-on' computer time be made available, and that other programming languages, in addition to BASIC, be included in the course content.

BIBLIOGRAPHY

Computer Programmer Aptitude Battery - Manual. Science Research Associates, Inc. (Developed by Jean Maier Palormo)

DODDS, W. References and Guide for the Computers 11 Course. Vancouver School Board, 1972.

MOODIE, A. G. An Evaluation of Student Experience with Computers in the Instructional Program of Two Secondary Schools of Vancouver, 1969-70. Research Report 70-13, Department of Research and Special Services, Vancouver School Board, June 30, 1970.

Recommendations Regarding Computers in High School Education. Conference Board of the Mathematical Sciences, Committee on Computer Education, Washington, D. C., April, 1972.

APPENDIX A

Appendix A, "References and Guide for the Computers 11 Course", has been included in copies sent to the Department of Education in Victoria, but has not been included in this copy of the report. However, copies are available upon request from:

The Department of Planning and Evaluation,
Vancouver School Board,
1595 West 10th Avenue,
Vancouver 9, B. C.

TABLE OF CONTENTS

TOPIC	PAGE
INTRODUCTION TO THE COURSE	2
LIST OF TOPICS BY UNIT	5
LIST OF BASIC'S FEATURES, BY UNIT	7
GUIDELINES AND REFERENCES, BY UNIT:	9 ff.

VANCOUVER SCHOOL BOARD

COMPUTERS 11

by: Mr. W. Dodds
Computer Consultant
Vancouver School Board
March 23, 1972

Overview

1) This course is for beginners to computers.
2) It presents conceptual introduction, through a broad overview of computer applications.
3) This overview would be useful to the student:
   a) As a voting citizen (invasion of privacy by computers, pollution control by computers, etc.)
   b) As a future employee whose job involves preparing input to, or using output from computers, or proposing new applications.
4) This course is not intended as vocational training, but does serve incidentally to introduce the student to the vocational and career possibilities in computer related employment.

Objectives

From greatest to least emphasis, the purpose is that the student gain a conceptual understanding, a little fundamental knowledge, but very limited skills, regarding the computer's:

1) application areas of social significance.
2) potential and limitations.
3) fundamentals of hardware and software.

Implementation

Experience has repeatedly shown that students learn best when they are interested. Also, true understanding and ability to apply what they have learned is best achieved by having students participate, in activities where transfer of knowledge and self-direction are implicit.

For these reasons, the students will actually write simple computer programs, and run them on the computer. Students find this very interesting and a great assistance to gaining true insight into abstract concepts. It is also relatively cheap, in dollars, now that "mini-computers" have been developed. Movies, lectures, quizes, and other non-laboratory oriented teaching techniques will also be used occasionally.

Explanation of this Booklet:

Objectives and a topic list are supplied on the next few pages. This list indicates the suggested number of computer programs per unit of study that an average student might code. Naturally, some students will likely code less, and some will code more. Encourage each student to work at his own speed, within each unit of study. However, the time limit in numbers of days for each unit should be followed as closely as possible, so as to give proper relative emphasis to each main area of application of computers in society.

The teacher is supplied with a few copies of this set of reference notes and guides. It is sufficiently detailed etc. to allow the brighter than average student to work ahead by self-study at his own speed, with little teacher assistance. However, average and slower students will require frequent, brief, explanatory class lessons by the teacher. To facilitate the planning of such lessons, a list of the BASIC language features introduced in each unit of study is supplied on the next few pages after the topic list.

Student Exercises:

Each student receives 1 booklet containing all the problems for the whole course. Answers to these problems are supplied in a few copies of a separate booklet, for occasional reference by the students. The main purpose of the answers is to help clarify the intent of each coding problem. Students should be frequently reminded that other variations of the solutions might be equally correct or even better.

In general, students will require several class periods and several re-runs on the computer, to code, test, and debug each program. Therefore, it is negative motivation to suggest that students try for "no errors" on their first computer run of any program. Students find such an objective very discouraging. Coding programs should be and usually is FUN for students. After the first few weeks, students will be able to start on the next problem while still debugging the previous problem, and thus more efficiently utilize their in-class time.

Varieties of BASIC used (software):

The BASIC language supplied by the latest and least expensive batch-mode "school computers" of several manufacturers contains all the language facilities needed by this course. However, the Vancouver School Board's "school computers" are of older vintage. Thus, the BASIC language facilities required are supplied by locally developed modifications to our older, limited version of BASIC. These modified versions are titled DISK BASIC and BUSINESS BASIC respectively. This course does not require MATRICES capability. Requirements are shown by a letter X in the following table (units #2,4,6,8 are theory only). Brackets means software is optional.

UNIT #	1	2	3	4	5	6	7	8	9	10
Regular BASIC	X						X		X	(X)
DISK BASIC			X		X					(X)
BUSINESS BASIC					X					(X)

Special Handling of Unit #10:

Instead of waiting until nearly the end of this course to have a chance to do computer programs of their own choice, most students would rather work throughout the year on their special interests. Thus, Unit 10 should be in the form of special projects upon which each student individually works during the inevitable spare time which will occur during the class time. Such spare time results when a student occasionally has very few errors to debug, computer breakdowns etc.

TOPIC OUTLINE, GENERAL

UNIT	TITLE	PERIODS OR DAYS (1 hour each)	PROGRAMS (coded by students)
1	CALCULATOR APPLICATIONS: - lengthy or repetitive calculations. - graph plotting from equations. - searches by "trial & error".	16	4
2	MAIN COMPONENTS OF COMPUTERS: - an overview of the purposes & relative speeds of the Central Processing Unit, and Input/Output devices hardware. - an overview of the purpose & nature of compilers and job control commands. - an overview of the purpose and types of data files organization.	1 1 2	- - -
3	CLERICAL APPLICATIONS: - data files methods of creation and maintenance (simple). - document printing (bills, paychecks, etc.)	20	4
4	SAFEGUARDS AGAINST COMPUTER ERRORS: - Central Processing Unit: - parity checks. - bit overflow detection. - internal codes and validation checks. - Input/Output Devices: - parity checks. - double read heads. - checksums, cyclic checks. - phase encoding. - File Organizations: - header/trailer labels. - expiry dates, passwords. - Operating Systems: - file access, checkpoint-restart. - Program Controls: - data types, value ranges. - External Controls: - user verification. - adding machine control. - backup copies, generations. - security against intruders.	5	-
5	DATA RETRIEVAL FOR DECISION MAKING APPLICATIONS: - Comprehensive report printing and presorting. - Exception reports. - On-line enquiry (simulated, only). - Management information systems (generalized report generators) ... theory,& supplied program used. - Data base files structure and uses(theory only).	35	6
6	JOB ROLES AND METHODS FOR COMPUTER PERSONNEL: - Operator, operations manager. - Programmers (application, systems), analysts. - user departments. - flow charting, documenting.	5	-

7	SIMULATION AND MODELS APPLICATIONS: - simulate behaviour of a physical object (thrown ball, teeter totter, swing, etc.). - make a simple model of a fictitious animal, economy, etc.	15	2
8	OPERATING SYSTEMS & SOFTWARE: - purpose. - services offered. - advantages/disadvantages. - comparison of "high-level" programming languages.	3	-
9	REAL TIME SYSTEMS APPLICATIONS: - process control (simulated). - conversational programming. - teleprocessing: purpose and applications (theory only).	5	1
10	PROGRAMMING APPLIED TO SCHOOL SUBJECTS: - students' own choice, from list of project suggestions, or teachers' choice.	17	1 to 3
	TOTALS	125	18 to 20

SEQUENCE OF INTRODUCING "BASIC'S" FEATURES

-The sequence is approximately according to increasing difficulty, as previously observed with students learning their first computer language.

-Units #2,4,6,8 do not appear below because they involve no programming.

UNIT #	PROBLEM #	FEATURES OF "BASIC" INTRODUCED :
1	1,2	-SCRATCH, END, LIST, RUN.
		-Arithmetic symbols + - * /
		-Constants and Variables (simple).
		-LET for data assignments and for calculation.
		-PRINT for one Variable (simple).
		-Statement numbering.
		-Optic card marking and sequence.
	3,4	-FOR ...NEXT using constants.
		-FOR ...NEXT to generate data.
		-PRINT with commas for spacing columns.
		-PRINT for titles above columns.
		-SQR,and use of parentheses ( ).
	5,6	-TAB using a variable (with PRINT).
		-Nested FOR ...NEXT.
	7,8,9	-IF ...THEN with constants.
		-Relational operators =, #, <, >, <=, >=, <>.
		-GO TO for causing looping.
		-LET used for counting.
		-PRINT with semicolons for close spacing.

UNIT #	PROBLEM #	
3	1	-READ and DATA statements.
		-Alpha data and printing of it.
		-Sequential creation of magnetic disk file.
		-Saving of magnetic disk file.
		-Sequential reading of magnetic disk file.

UNIT #	PROBLEM #	
3	2	-Updating file via direct rewrite. -Extending file via set to end-of-file and sequential write.
	3	-Detecting end-of-file via signal numbers, etc.
	4,5	-Direct read of magnetic disk files for table lookup. -INT (integer) built-in function. -PRINT blank lines.

UNIT #	PROBLEM #	
5	1	-Character strings via dollar $ Variables, using READ, DATA, PRINT, and IF ...THEN . -Special RUN number need by Business BASIC.
	2	-PRINT using format statements. -Integer arithmetic with non-automatic decimal point.
	3,4	-CALL (11), CALL(17). -AND, OR logic with IF ...THEN
	5	-Password protected files. -Copying file to file.
	6,7	-Subscripted variables. -INPUT instead of READ, for hands-on mode.

UNIT #	PROBLEM #	
7	1,2	-SIN, COS.

UNIT #	PROBLEM #	
9	1	-GOSUB, RND (random number generator). SGN (test the +,Ø,- status).

UNIT #	PROBLEM #	
10	1,2,3	-Any of the built-in functions, MATRICES, or students' choice of previously learned BASIC facilities. Exponential number notation.

REFERENCE & GUIDE — UNIT # 1

HOW TO TELL THE COMPUTER WHAT YOU WANT IT TO DO:

Before starting exercise #1 of this course, the beginner to computers should spend 15 to 30 minutes learning the following material. The following page references are from the textbook titled "A GUIDE TO HP EDUCATIONAL BASIC", which can be ordered (catalog # 105) from your VSB Hewlett-Packard Computer centre (5 copies per classroom). The material on these referenced pages below can be either read by the student, or presented by the teacher to the whole class. The sequence of the pages referenced has been found most suitable, in this course author's experience, as shown below:

PAGES: 1-1, 1-10 to 1-16, 1-27, 1-28, 1-36 to 1-40, 2-11.

HOW TO PENCIL MARK YOUR COMPUTER PROGRAM ONTO INPUT CARDS:

In the same reference book, read pages: A-1 to A-10; and 7-9 item # 1.

The above information should not be covered until the student has solved problem # 1, because he has no need to mark cards until then, & might be overwhelmed.

GENERAL REFERENCE SHEETS:

Available to each student (one copy per student) are several summary reference sheets, which can be ordered as catalog #'s 102, 103, 104, plus several others when the student has become more experienced.

REVIEW

When you have finished unit # 1 coding problems (you might not have done them all, since there are more than the average student would finish), try the review questions on the next page.

REVIEW QUESTIONS ON "BASIC"

1) If you want to execute a BASIC program, what must be the first card and what may be the last card in your deck ?

2) What is the name of the BASIC statement which must always have the largest statement number in the program, and what is a good statement number to use, to guarantee it will be the largest ?

3) How would you code a statement number 10 which would tell the computer:
   a) to print the letter Y ?
   b) to print the numeric value of the variable X ?
   c) to print the numeric values of A, B, C, D, E all on the same line ?
   d) to print the values of A, B, C, D, E, F, G all on the same line ?
   e) to print the letters TOTALS starting at column 34 on the line ?
   f) to print the value of Y at column X on the line ?

4) What are the two symbols either one of which can cause the computer to stay on the same line it has just printed, instead of spacing down to a new line after printing ? Where do these symbols go, in the PRINT statement ?

5) If you wish to give the variable R the value of 5, how would you code this, as statement number 23 ?

6) How can you tell the computer to print the result of the calculation A multiplied by B, if you only wish to code one statement, number 35 ?

7) The area of a circle is given by multiplying Pi times the radius squared, where Pi has the value of 3.14159. What are two ways you could code this calculation, in BASIC ?

8) Code one statement, number 48, which will calculate the average of the three numbers A, B, C (this is done by adding the three numbers first, and then dividing by 3).

9) When you ask the computer a question by using the IF ... THEN statement, if the answer turns out to be "no", which statement number will be executed next (it is either the number after THEN or the next largest number compared to the number in front of IF) ?

10) Code one statement which will read one value of A and one of B. Also code one statement which will supply the data values for A as 2, 7, 9; and B as 8, 3, 5 . Use statement numbers 25 and 27.

ANSWERS: 1) SCRATCH and RUN 2) 9999 END 3)a) 10 PRINT "Y"
3)b) 10 PRINT X c) 10 PRINT A,B,C,D,E or 10 PRINT A;B;C;D;E
3)d) 10 PRINT A;B;C;D;E;F;G e) 10 PRINT TAB(34) "TOTALS"
3)f) 10 PRINT TAB(X) Y 4) at the end of PRINT statement, a semicolon or a comma if less than 5 items on line. 5) 23 LET R = 5
6) 35 PRINT A * B 7) 40 LET P = 3.14159 and 42 LET A = P * R * R or else
40 LET A = 3.14159 * R ↓ 2 or etc. 8) 48 LET M = (A + B + C) / 3
9) Next largest number compared to statement number in front of IF .
10) 25 READ A,B and 27 DATA 2,8,7,3,9,5

REFERENCE & GUIDE **UNIT # 3**

CREATION & MAINTENANCE OF DATA FILES:

A magnetic disk file is usually only created once, but it is never in a finished state, and is maintained (kept up to date)by deleting, adding, and changing data records. It is usually created so that the records are in an increasing sequence, according to the key field, which is usually the first field. Thus, if the key field was the surname, then JONES' record would precede that of SMITH.

Maintenance can be done by either of two methods (or a mixture of both methods, with certain limitations). Direct-write updating is fastest, if less than 15 % (approximately) of the data records are to be affected. However, for larger percentages, the time required for the mechanical motion of the access mechanism on a disk device adds up to more time than would be required to use the other method, namely, sequential update. In the sequential method, every record in the master disk file must be read, but the time required to read one record is considerably less than that required in the direct method. However, in the direct method, only the affected records need to be read/written (eg; 15 %).

Thus, time required is the chief concern in choosing which method to use. But another consideration is that direct updating is easier to code computer programs for than is sequential updating. Also, direct updating allows the modifications to be made in chronological order, whereas sequential updating requires that the updates occur in some alphabetical or numeric sequence. For example, if several customers want to buy several items, direct updating would allow their requests to be handled on a "first come first serve" basis. Sequential updating might not be able to be quite so fair to the customers. For sequential updating, the modifications must be pre-sorted into the same sequence as the master file that is to be updated. Problems # 2 and 5 are examples of direct and sequential updating, and will help clarify these two ideas better than any explanation here could possibly do(however, no actual modification of data records occurs in #5).

Similarly, when utilizing disk files, either direct or sequential retrieval of the data records can be used. Again, the same relative advantages and disadvantages occur as explained above for updating. Problems #4 and 5 are examples of direct and sequential retrieval of information, respectively, where two or more files are involved. Problem #1 is a trivial example of sequential retrieval, since only one data file is involved.

DISCUSSION OF PROGRAMMING PROBLEMS:

#1) One copy per student is available of the reference sheet titled:
& 2) "HOW TO USE ALPHA DATA ... FOR HP BASIC" (order as catalog # 106). It also contains an example program of how a disk file can be created and later listed on the printer (involves a mixture of alpha data & numeric data).

Also available:"DISK FILES FOR ED. BASIC"(15 pages) as catalog # 202, and "DISK FILES BASIC ... SUMMARY REFERENCE SHEET"(catlg.#201)-rules & examples.

#4) To truncate pay to 2 dec.places, multiply by 1ØØ, use INT(P), & divide by 1ØØ. To direct-read disk file;example,reference sheet (catalog # 201). This problem can be temporarily simplified by not truncating the pay to two decimal places, and by not rounding up to the nearest penny. One space will always occur between $ sign and first digit of the pay. A forger could put an extra digit in the space, & thus illegally increase the value of the paycheck. To prevent this, make pay negative (a minus sign - goes into space). To simplify, do not check for the possible error "no such employee # on file".

#5) Advanced students might load data cards onto disk file#3, sort into file#4, and read files #1 & 4, to print checks (method used by real programmers).

TEACHER SUPPLIED PROGRAM TO CREATE DISK FILE FOR PROBLEMS # 4, 5

```
SCRATCH   TEACHER NAME
1Ø READ A,B,C,D,E,F,G,H,I
2Ø CALL (33,2ØØ1)
3Ø CALL (9,1,3,A,B,C,D,E,F,G,H,I)
4Ø CALL (22,2ØØ1)
5Ø GO TO 1Ø
6Ø DATA 1Ø47,1Ø,19,13,9,2Ø,8,15,2.43
7Ø DATA 1273,18,3,12,1,18,11,17,2.67
8Ø DATA 1436,2,12,15,3,11,5,13,1.85
9Ø DATA 2584,11,1,4,1,13,19,14,2.18
1ØØ DATA 2739,4,12,5,23,9,19,12,2.31
11Ø DATA 2894,19,2,12,1,11,5,16,2.28
12Ø DATA 2951,23,13,1,7,5,5,13,2.54
13Ø DATA 3125,6,5,12,12,9,19,14,2.67
14Ø DATA 3257,3,12,1,22,1,12,12,2.31
15Ø DATA 3742,13,4,15,18,14,5,17,1.85
16Ø DATA 4375,8,7,18,5,5,18,15,2.54
17Ø DATA 4859,2,13,15,15,18,5,16,2.43
18Ø DATA 5137,16,1,18,14,5,2Ø,13,2.28
9999 END
LIST
RUN    CREATE UNIT # 3
```

REFERENCE & GUIDE **UNIT # 5**

COMPREHENSIVE REPORT PRINTING, AND PRE-SORTING:

The topics in this unit are taken in the same sequence as the various methods of report printing were invented, historically speaking. Thus, in the early days of computers, mainly comprehensive style reports were printed, in business, industry, and government offices using computers. A comprehensive report is simply a list of every record of information there is in a data file. Usually, it is one line of printing for each data record, and usually not all the data fields are printed. Those data fields that are printed are always the same respective fields, for each record. Programming problems # 1 and 2 show what such reports look like.

At first, people were very impressed by the tremendous speed and huge volume of pages that the computer produced. For example, a file of several thousand customers would result in a 150 page report, taking 10 or 20 minutes to print. A company maybe only wanted to use the report to find out which of their customers had not paid the latest bills owing. This might only be a few hundred people, but unfortunately, someone had to search through this whole report of several thousand people, reading each line, to find the delinquent debts. Thus, the exception report was invented. It only prints those people with the bad debts, or whatever other qualification is used to select them from the whole file

Nevertheless, comprehensive reports still have desirable uses, today. Often, a comprehensive report (one) is less expensive than many exception reports, each for a different purpose, provided you do not have to use the comprehensive report very often. Also, sometimes a complete listing of a file is required. For example, a school list of all the students and their home phone numbers would not be too useful if many students were not printed.

The same data file can be sorted into two or more different sequences, and each sequence used to print a separate comprehensive report. Although both reports might contain the same information, each would be in the easiest to use order for people looking up what they need to know. Thus, in problems #1 and 2, the two reports allow a person to look up any city, to find out who the salesmen are; or to look up any salesman, to find out his relative $ sold. Surveys have shown that up to 60 % of computer time is spent sorting data files (for various reasons), traditionally. Thus, programmers do not ever code a sort program, because such programs are so important & frequent that the computer manufacturer usually supplies a highly efficient, general purpose sort program. Exceptions would be certain scientific applications.

DISCUSSION OF PROGRAMMING PROBLEMS # 1, 2 : References: catalog items # 301 302

#1) It is much simpler to code this program so that it prints <u>every</u> city name, and also so it prints the <u>cents</u> (but not the decimal point). Once the programming is debugged and running, <u>then</u> code further statements to achieve the requirements temporarily omitted.

It is also easier <u>not</u> to round up to the nearest dollar, before dropping off the cents. Thus, this program can be simplified, if the student desires.

#2) A better than average student might pre-sort by supplying appropriate coding in his program, instead of manually pre-sorting as suggested. If so, this is done by reading the first data card and saving it (use LET). Then, all other data cards are read. Each time a bigger # is found, it is saved in place of the previous biggest #. At last record, use RESTORE, and repeat for next line Good students should try rounding up the pennies, before dropping decimals.

TEACHER SUPPLIED PROGRAM TO CREATE DISK FILE FOR PROBLEMS # 3,4,5,6,7

```
SCRATCH   TEACHER NAME
1Ø READ A,B,C,D,E,F,G,H,I,J
2Ø CALL (33,2ØØ1)
3Ø CALL (1Ø,1,3,A,B,C,D,E,F,G,H,I,J)
4Ø CALL (22,2ØØ1)
5Ø GO TO 1Ø
6Ø DATA 13527,1,28,1,3,8875,1ØØØØ,324,1,1
7Ø DATA 21784,Ø,27,-1,-2,11538,2ØØØØ,158,-1,Ø
8Ø DATA 27349,-1,62,1,Ø,15ØØØ,5ØØØØ,452,-1,-1
9Ø DATA 35294,1,43,Ø,Ø,94ØØ,5ØØØ,287,-1,Ø
1ØØ DATA 38274,Ø,32,1,2,1258Ø,3ØØØØ,835,1,1
11Ø DATA 43857,1,26,1,1,82ØØ,15ØØØ,427,1,Ø
12Ø DATA 52953,1,39,-1,-2,1472Ø,35ØØØ,628,-1,-1
13Ø DATA 55836,Ø,57,-1,Ø,72ØØ,25ØØØ,573,Ø,Ø
14Ø DATA 57249,-1,58,Ø,Ø,187ØØ,2ØØØ,936,1,Ø
15Ø DATA 64785,1,34,1,Ø,115ØØ,1ØØØØ,794,1,1
16Ø DATA 67492,1,29,1,3,1346Ø,2ØØØØ,158,1,Ø
17Ø DATA 74962,-1,47,-1,-5,1248Ø,15ØØØ,452,Ø,Ø
18Ø DATA 83579,Ø,58,1,Ø,925ØØ,3ØØØØ,324,Ø,-1
19Ø DATA 87325,Ø,53,Ø,Ø,1Ø735,5ØØØ,287,1,Ø
2ØØ DATA 94738,1,36,1,-2,1274Ø,25ØØØ,452,1,1
9999 END
LIST
RUN    CREATE UNIT # 5
```

TEACHER SUPPLIED "GENERALIZED REPORT GENERATOR" FOR PROBLEM # 7

```
SCRATCH   TEACHER NAME/GRG#5-7
6 DIM X(11,3),P(11),F(1Ø)
7 LET C = -1
8 LET Q = 9E37
9 PRINT
1Ø FOR N = 1 TO 11
2Ø READ X(N,1),X(N,2),X(N,3)
25 LET C = C + 1
3Ø IF X(N,1) = Ø THEN 45
4Ø NEXT N
45 LET R = Ø
5Ø FOR K = 1 TO 11
6Ø LET P(K) = Ø
7Ø NEXT K
8Ø CALL (1Ø,1,1,F(1),F(2),F(3),F(4),F(5),F(6),F(7),F(8),F(9),F(1Ø))
9Ø FOR N = 1 TO 11
1ØØ IF X(N,1) = Ø THEN 18Ø
11Ø IF X(N,2) = -1 AND F(X(N,1)) < X(N,3) THEN 15Ø
12Ø IF X(N,2) = Ø AND F(X(N,1)) = X(N,3) THEN 15Ø
13Ø IF X(N,2) = 1 AND F(X(N,1)) > X(N,3) THEN 15Ø
14Ø GO TO 16Ø
15Ø LET P(N) = F(X(N,1))
155 LET R = R + 1
16Ø NEXT N
17Ø GO TO 185
18Ø LET P(N) = Q
185 IF R<C THEN 2ØØ
19Ø PRINT TAB(5);P(1);P(2);P(3);P(4);P(5);P(6);P(7);P(8);P(9);P(1Ø);P(11)
2ØØ GO TO 45
9999 END
```

EXCEPTION REPORT PRINTING, AND ON-LINE ENQUIRY:

Exceptions reports are tailored for a specific purpose, and are thus easier to use than one comprehensive report manually searched for various purposes. An example is the report printed in problem # 3. Sometimes an exception report is just a simple selection of certain records from a large master file (as in problems # 3, 4). However, before printing the report, the selected records often are sorted into a sequence which makes the exception report easier to use. Thus, in problem # 3, a particular salesman # will appear on widely scattered lines, throughout the report. It would be much nicer if all the selected customers for a particular salesman appeared on consecutive lines of the report. Then, we could cut the paper into pieces, one per salesman, and give it to the salesman to help him do his job better. Notice that this is sorting after the records have been selected, and is more efficient than sorting the much larger number of master file records, before selection(time-wise).

While exception reports are faster to use than comprehensive reports, they are still too slow and awkward in cases where many people are looking up things very frequently. Consequently, on-line enquiry was invented. This prints a different kind of report, than exception reporting. It does not search through the whole file to seek out all records that match certain qualifications (such as all men earning more than $10,000 yearly). Instead, it simply allows a person to request the record for a particular man etc., whether or not he matches any selective qualification. Thus, a manager simply enquires how much Joe Blo is making, rather than requesting an exception report containing many salesmen. Usually, the question and the answer are done via a special typewriter attached to the remote computer through telephone lines, etc. The answer may simply be a copy of the whole data record on Joe Blo, including many things the manager is not presently interested in. Clearly, on-line enquiry is both better and worse than exception reporting, depending upon your purpose & needs. It is used today in banks (to know your balance), retail stores (to certify your credit rating), airlines reservation offices, inventory warehouses(to see if stock is on the shelf to fill a customer's order), police patrols (to check out suspicious persons against the "wanted" list), telephone & electric utilities (to clarify complaints about bills), etc, etc. Usually, on-line enquiry also has the ability to modify the contents of a data record in the master file. Problems # 5,6 are examples of on-line enquiry, with modification capability. Besides speed, an advantage of on-line enquiry is accuracy, compared to a report which may have been printed hours or days previous (and thus not be up to date regarding recent modifications to the master file).

MANAGEMENT INFORMATION SYSTEMS & GENERALIZED REPORT GENERATORS:

This is a more ideal situation than above, because it combines both the on-line enquiry and the exception report capabilities; plus eliminating the need to have computer programs written for each exception report required. This is both far cheaper and far faster than having to wait several days, weeks, months, for a programmer to code what you need. It is still a tremendous advantage compared to the above discussions, when only batch mode (not on-line enquiry) is allowed using a generalized report generator. Problem # 7 is an example of a generalized report generator that can be used either in the batch or the on-line enquiry mode. Professional versions of generalized report generators can be purchased for prices ranging from $5,000 to $40,000 and up. They have many capabilities, such as pre-sorting, post-sorting, comprehensive & exception reporting, and printing of columns of figures which are not in the data file but which can be calculated from information in the data file, calculating and printing column subtotals, and totals, printing columns in any position on the page (left to right) with any titles you wish, and so on. Complex or specialized work such as printing paychecks, etc. still requires the coding of programs, however.

DISCUSSION OF PROGRAMMING PROBLEM # 7

Students may wish to try the teacher supplied generalized report generator to print other reports, than requested in problem # 7, such as one or more of the following reports (or a report of their own invention):

-all customers who have poor health (or top health).
-all customers who are under 25 (or over 65, etc.)
-all single customers (or all married customers, etc.)
-all married customers without children(or with them, etc).
-and so on, for the other data fields of each data record.
-any combinations of the above requirements.

Notice that this version of the supplied program only handles selections based upon one or more qualifications connected with AND (you cannot use OR). However, by removing statement # 185, the program will only handle qualifications which are connected with OR (you cannot use AND). Thus, version #1 of program handles things such as:
all customers with insured wives AND age over 50.

Version #2 handles: either poor health OR over 65 years age customers.

Neither version handles: (top health AND under 25) OR(average health AND under 35

Of course, professional versions would handle all the above and more.

DATA BASE FILES STRUCTURE AND USES:

These have come into being partly because of the above modern capabilities, and partly because of the high cost of maintaining and modifying programs and data files. Two characteristics distinguish Data Base File Systems from the more traditional data file systems. Firstly, there is a tendency to have only one master file, instead of many independent files, for a specific set of data. Secondly, the number of fields per data record can be increased, decreased, or arranged in a different relative position, without having to change any of the computer programs which use that file. However, in the traditional situation, a company might have several different master files, for different purposes, all containing partly the same data, in slightly different format. Thus, each employee's name and address might be in the separate data files of the following departments of a company: personnel (history, etc.) dept., payroll dept., sales dept. (for manager to keep track of his workers' performance). If the employee changes address/name (quits, marries, moves, transfers), there are 3 files to update. Besides being less expensive, one file instead of three would be more accurate, because with 3 files, a person can not be sure which file is the most up to date, in case of discrepancies. As for changing programs when the number or position of data fields is changed, this is easy to understand if you think about the following statements for a computer program:

1Ø READ A,B,C or else 1Ø READ B,A,C or else 1Ø READ A,B,C,D

Certainly, each statement will accomplish a different thing than the other two. As the data fields are re-arranged, or added to a data record, the first, second, or third of the above READ statements would be required (a change in the computer program). Usually, companies have hundreds of programs, and in each one, many changes would be required, taking hundreds of hours costing thousands of dollars.

The Data Base system has only become possible recently, and perhaps mainly for two reasons. First, computer software has only recently been developed & proven for making possible the above explained independence from data fields in records. Secondly, business, government, & industry have only recently become sufficiently in favor of computerized data processing to be willing to use such a system. However, it is now coming into common use, & is even available on small computers

REFERENCE & GUIDE UNIT # 7

SIMULATES & MODELS:

The difference between the meaning and usage of these two words is somewhat arbitrary and hazy, for modern digital computers. However, a simple definition may help:

Simulate programs cause the computer to behave in a similar or analogous manner to some real, physical object. For example, a picture of an airplane can be shown on a Television style picture tube, called cathode ray tubes, or CRT's. The computer can produce and control this picture so that the airplane appears to behave under the various specified conditions just the same as would a real airplane. This is much cheaper, safer, faster, and easier to use than a real airplane, for doing experiments to find out what would happen in various situations (eg; high winds,etc).

Model programs cause the computer to predict what would happen to some physical or abstract thing, under various specified conditions. Usually, the answers (predictions) are output in the form of numbers rather than pictures, graphs, etc. For example, a mathematical model of the atmospheric weather for the world would predict the weather at any future time at any place in the world, provided the initial conditions were supplied (eg; today's weather at various key places throughout the world). Such a model would also be very useful for experiments such as predicting what would happen to the future weather if the atmosphere were polluted in some specified manner.

It was once common to use ANALOG computers for doing simulations, and DIGITAL computers for doing models. Analog computers do not output numbers, but instead output continuously varying voltages (to control physical objects, plotters, CRT's, etc.). However, today's digital computers are so commonly available and relatively inexpensive as to practically have replaced analog computers, for simulation applications. If a digital computer is used, then math. calculations which output numbers are used for both simulation and for model programs. Thus, the difference is now more hazy than it used to be, for these two terms.

DISCUSSION OF PROGRAMMING PROBLEMS # 1, 2:

#1) By observing the simulated behaviour of the golf ball, we can learn many things. We already know that the distance along the ground is 10 yards, between each pair of asterisks printed by the computer. Also, this means that the scale of the output diagram is 90 feet/inch.

The computer is used for models & simulations because it is not practical to do the great many calculations required, by other ways.

Thus, use your computer output diagram to answer the following questions:

- Which angle of hit makes the golf ball go the farthest?
- How many yards did each of the 3 golf balls travel?
- What maximum height in feet did each golf ball reach?

The accuracy of the diagram is rather low. However, it is improved somewhat if your program rounded UP each value of H before the TAB automatically truncates H to an integer, when the asterisk is printed.

If time is available, your program could be modified to make a game in which the player specifies the initial speed and angle of hit. The purpose is to have the ball fall into a hole 250 yards away. To make the game more realistic, your program could use the random number generator RND to cause the golf ball to roll an unpredictable distance, after landing, of between 5 and 45 yards (by dividing the random number by 2), from random #'s of 10 to 99.

#2) Good accuracy is not necessarily obtained by using simulations and models. It depends on the quality of the math calculations. For example, our model did not take into account such factors as air friction, spin on the ball, rain, wind, elevation above sea level, and so on. However, the computer output can still be useful, depending upon our needs, and assuming we are aware of the above limitations. Of course, the fact that the computer did the calculations does not somehow magically make the predictions valid... as sometimes implied in news stories.

In spite of the above inaccuracies, our model probably allows you to answer the following question:

-What is the best angle of hit, to get the greatest distance while the golf ball is in the air?

Our model gives us more accurate answers more quickly than does our simulation program, for this kind of problem. If you have time, use your model program to answer the following questions:

-How fast an initial speed would you have to hit the golf ball if it is to travel 300 yards while in the air, assuming the best angle of hit was used?

-At what angle should a warship aim its guns if the shell speed is always initially 1000 miles per hour(1430 feet/second) and the target is 10 miles distance (52,800 feet or 17,600 yards)?

OTHER APPLICATIONS OF MODELS & SIMULATIONS

Attempts are continually being made to use computer with models of a country's Economy (to predict the effect of government policies), the behaviour of humans in various psychological situations, etc., etc. Thus, the physical world & machines are not the only uses of models.

PROCESS CONTROL BY USING COMPUTERS:

Computers have been used to control the operation of machinery almost since the computer was invented. For example, computers are used to control such things as oil refineries, aluminum smelters, pulp & paper manufacture, some parts of lumber manufacture (sawmills), distribution of electricity to homes and industries, telephone equipment in central exchanges, welding machines, metal working machines (lathes, drills, etc.), rocket ships, sea going ships such as the super-tankers, guns on ships and airplanes, draughting machines, and so on. Usually the computer is not in complete control. Instead, a human being is on hand to monitor the computer's actions, to ensure that no huge, costly blunder is occurring. Also, not all aspects of a manufacturing, refining, etc. operation can be effectively controlled by computers.

Generally, most computer control has the following method: the computer reads temperatures, pressures, physical object sizes, flow of liquids, etc. through wires connected to special measuring devices (voltmeters, for example). Then the computer compares these measured values against what they should be. As a result, the computer sends signals to mechanical and electrical devices so as to adjust the temperature, pressure, etc. to what it should be. Finally, the computer checks to see if its efforts have resulted in correct re-adjustment being achieved (by repeating the above measurements, and other above procedures indefinitely). This self-checking of the results of its actions is called feedback control, and is perhaps the most important aspect for success.

DISCUSSION OF PROGRAMMING PROBLEM # 1:

#1) Here, we are only pretending to control real machinery. Actually, there are no wires going from the computer to the machinery (which is also imaginary). Nevertheless, the feedback control is very much in effect for this problem. Thus, when the student's program increases the fuel for heating the chemicals, there will be a corresponding increase in the measured temperature which the program next "reads", with chance fluctuations.

Of course, such a simple control could be achieved by a non-computer device, such as the common household thermostat. However, the computer is required to control this process in the chemical factory because the continuous addition of the fraction temperature/pressure is required so as to know when the chemical process has been completed.

The measurements of the chemical process are simulated by the following teacher-supplied subroutine that each student must code into his own program (it is common for different parts of a computer program to be coded by different people, depending upon complexity, special skills, etc):

```
9000 REM ...'MEASURES' T, P
9010 IF D = 0 THEN 9050
9020 LET R = V = D = 0
9030 LET P = 144
9040 LET T = 156
9050 IF INT(P) # 144 THEN 9070
9060 LET Z1 = SGN(5 - INT(10 * RND(0)))
9070 IF INT(T) # 156 THEN 9090
9080 LET Z2 = SGN(5 - INT(10 * RND(0)))
9090 LET P = INT(P + R - Z1 * INT(2 * RND(0)))
9100 LET T = INT(T + V - Z2 * INT(2 * RND(0)))
9110 RETURN
9999 END
```

Note: the student should not see nor use this subroutine until he has coded and tested and debugged his own program. This is done by temporarily using a READ instead of the suggested GOSUB for obtaining readings of temperature and pressure, and by his supplying DATA which tests all possibilities.

PROGRAMMING PROBLEMS FOR "COMPUTERS 11" COURSE

UNIT 1

Example 1-1: Code a generalized program which will calculate the volume of any sphere, and print the answer. Use one LET statement to supply a value of 23.754 for R, the radius. Use another LET statement to calculate V, the volume. Use a PRINT statement to print the answer, V. The equation is $V = 4/3\pi R^3$. Use 3.14159 for π. Be sure to LIST your program before it is executed (RUN).

Solution 1-1:

```
   SCRATCH TEACHER NAME
1Ø LET R=23.754
2Ø LET V=4/3*3.14159*R↑3
3Ø PRINT V
9999 END
   LIST
   RUN STUDENT NAME
```

Problems:

1. Code and test a generalized program to calculate the area A of any circle of radius R, and print the answer. The equation is $A = \pi R^2$ where π is 3.14159. Use a LET statement to supply any value of R you wish.

2. Code and test a generalized program to calculate the volume V of a rectangular box, and print the answer. The equation is V = LWH where L is length, W is width, H is height. Use LET statements to supply any values you wish, for L, W, H.

Example 1-2: Modify the program from example 1-1, so that it will print both R and V, by using a different PRINT statement instead. Also, use the FOR and NEXT statements to supply R with the values of 1, 2, 3, . . ., 10; instead of using the LET statement to supply R with the value of 23.754. Thus, your modified program should now print two columns of numbers: one column for R values, one column for calculated V results. Also print the words RADIUS and VOLUME as titles above each column of numbers.

Solution 1-2:

```
   SCRATCH TEACHER NAME
1Ø PRINT "RADIUS","VOLUME"
2Ø FOR R=1 TO 1Ø
3Ø LET V=4/3*3.14159*R↑3
4Ø PRINT R,V
5Ø NEXT R
9999 END
   LIST
   RUN STUDENT NAME
```

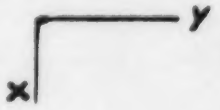
Problems:

3. Code a program to calculate the perimeters P and Q of a circle and of a square if each has the same area A of 1, 2, 3, . . ., 8.

The equations are $P = 2\sqrt{A\pi}$ and $Q = 4\sqrt{A}$. Note that square root is obtained by raising to the 0.5 power, or by using the built in function SQR. Print the titles AREA , CIRCLE P. , SQUARE P. , and three columns of numbers. Recall that π is 3.14159.

4. Code a program to calculate and print the surface area A of spheres which have a radius R of 2, 5, 8, . ., 17 respectively. Also print the value of R, and print titles above each column of RADIUS and SURFACE. The equation is $A = 4\pi R^2$.

Example 1-3: Code a program which will print a graph of the equation Y = 2X + 5, where X has the values 1, 2, 3, . . ., 12. The graph should have the axes as shown here:
Use FOR . . . NEXT to supply the X values. Use LET for calculating Y. Use TAB with PRINT, for plotting the points. Do not print the axes nor titles, etc. Print an asterisk * for each point plotted.

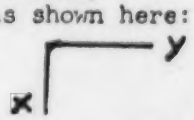


Solution 1-3:

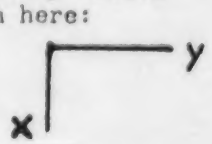
```
   SCRATCH TEACHER NAME
1Ø FOR X=1 TO 12
2Ø LET Y=2*X+5
3Ø PRINT TAB(Y) "*"
4Ø NEXT X
9999 END
   LIST
   RUN STUDENT NAME
```

Problems:

5. Code a program which will print 3 graphs of the equation Y = MX + 3, where X has the values of 1, 2, 3, . . ., 10 and M has the fixed values of 0.5, 1.0, 1.5. Do no print any titles or axes. The graphs should have the axes as shown here: Use asterisks for plotting all 3 graphs.



6. Code a program which will print 3 graphs of the equation Y = 2X + B where X has the values of 1, 2, 3, . . ., 10 and B has the fixed values of 2, 10, 18. Do not print any titles or axes. The graphs should have the axes as shown here: Use asterisks for plotting all 3 graphs.



Example 1-4: Code a program to calculate the cube of the numbers 10, 11, 12, . . . etc. and print each answer X on the same line, close spaced. However, when the answer X becomes bigger than 5,000, print the message FINISHED, and stop the execution so no further calculations nor answers are produced.

Solution 1-4:

```
   SCRATCH TEACHER NAME
1Ø LET N=1Ø
2Ø LET X=N*N*N
3Ø IF X>5ØØØ THEN 7Ø
4Ø PRINT X;
5Ø LET N=N+1
6Ø GO TO 1Ø
70 PRINT "FINISHED"
9999 END
   LIST
   RUN STUDENT NAME
```

Problems:

7. Code a program to calculate the interest earned at the end of each year, if the original bank balance B was $1,000, the interest rate R is 7% per year, and the interest earned I is always added to the bank balance B. Print the new balance, at end of each year, on the same line as previous balances. Stop when the balance B is bigger than $2,000, and print the message DONE. Use the equations:
   Interest = Old Balance x Rate/100
   New Balance = Old Balance + Interest

8. The half life of a radioactive material is 1 year. This means that one half of it disappears after 1 year, and the next year half of what was left disappears, and so on. If there was 5 lbs. to start with, calculate and print on the same line how much is left at the end of each year. Stop when less than 0.1 lb. is left.

9. A pair of animals from the rodent family gives birth to an average of 4 young, every 3 months. Half are male, half female, on the average. Assume that after 3 months, the young have matured, and they mate to produce offspring, as above. Also, the original parents mate at the same time. Calculate and print on the same line the total number of rodents, at the end of each 3 months, assuming none die. Stop when this total number of animals exceeds 1,000.

PROGRAMMING PROBLEMS FOR "COMPUTERS 11" COURSE

UNIT 3

Example 3-1: Code a program that will read a set of data cards, and create (write) a sequential "magnetic disk" file containing the data. Also code a program which will sequentially read the disk file, and print the data in each record, using one print line per record. Use file #2. The data consists of the weight, height, age, and identity # of people measured during an experiment, and is as follows: 165, 5.8, 25, 1435
135, 5.6, 28, 1278
180, 5.9, 26, 1649

Solution 3-1:

```
SCRATCH TEACHER NAME
1Ø CALL (48)
2Ø READ A, B, C, D
3Ø CALL (4, 2, 3, A, B, C, D)
4Ø GO TO 1Ø
9999 END
LIST
RUN STUDENT NAME
SCRATCH TEACHER NAME
1Ø CALL (4, 2, 1, W, X, Y, Z)
2Ø PRINT W, X, Y, Z
3Ø GO TO 1Ø
9999 END
LIST
RUN STUDENT NAME
```

```
5Ø DATA 165, 5.8, 25, 1435
6Ø DATA 135, 5.6, 28, 1278
7Ø DATA 180, 5.9, 26, 1649
```

Problems:

1. Code a program that will read a set of data cards, and create a sequential magnetic disk file containing the data. Also code a program which will sequentially read the disk file, and print the data in each record, using one print line per record. Use file #4. The data consists of the identity #, nickname, and weight of some experimental animals, and is as follows:

   327, JACK, 15
   254, JILL, 13
   436, JOHN, 14
   999, ENDS, Ø

   Use the "Alpha Data" capability for coding the nicknames into the DATA cards, and also for printing the nicknames in the second program (no need to "pack" the alpha data).

2. Code an update program to be run between your above 2 programs (file creation and file printing). It is to read data cards, and re-write the corresponding record on the disk file. If no such disk file record exists (considering the animals' identity #'s only), then the data record is to be added onto the end of your disk file.

The update data is: 254, JILL, 12
173, BILL, 13
436, JOAN, 14

3. Change all of your 3 above programs, and add a final end of data record to the data for question #2, so that your programs will not cancel with error messages #56 and 83. This is achieved by testing for the 999, animal identity #, and branching to end of your program when the 999 is detected.

4) Code a program which will print paychecks with the following format (including the *'s): **The disk file used is password protected.**

****

THE WHAT CO. LTD.

SEPT. 24/72.

PAY $XXX.XX

TO X XXXXX

THE UNKNOWN BANK
WHICH CITY, B.C.

****

Note that the pay is rounded to two decimal places. The name, total deductions, and hourly rate of pay for each employee are obtained from disk file #1 (supplied by the teacher). Each disk record has the following format, with employee # as the key field:

Employee #	Initial	Surname	Deductions	Hourly rate
	X	X X X X X		
(numeric data)	(alpha data) (not packed)		(numeric data)	(numeric data)

DATA cards made up by the student contain the following information:

Employee #	Regular Hours	Overtime Hours
1436	40	3
3257	40	6
2894	40	0

Overtime pays 1.5 times regular pay. Use a direct read to get the necessary information from the disk file. **No error checking needed.**

5) Re-arrange your DATA cards for question #4 so that they are in ascending order by employee #, and add one extra DATA card for employee # 3Ø25, who worked 21 hours regular and no overtime. Also change your question #4 program so that it uses sequential read for disk file #1. There are more employees on the disk file than there are in the DATA cards, and the disk file has no such employee # as 3Ø25. Thus, also supply extra coding to test if the employee # read from the disk file is smaller than the employee # read from the DATA card. If it is, keep reading disk records but not DATA cards, until employee #'s match (equal), and then print paycheck as usual. If the disk file employee # is greater than the DATA employee #, then print the error message NO SUCH EMPLOYEE # AS NNNN, read another DATA record, and go back to the beginning of your extra coding for testing which employee # is bigger, etc. The disk file is in ascending order by employee #. Print one paycheck for each valid DATA card employee #.

PROGRAMMING PROBLEMS — UNIT # 5

1) Code a program to print a report of each city's sales results in $, as shown below:

CITY	SALESMAN	DOLLARS SOLD
XXXXX	XXXXX	XXXX
	XXXXX	XXXX
XXXXX	XXXXX	XXXX
XXXXX	XXXXX	XXXXX
	XXXXX	XXX
	XXXXX	XXXX

Notice that when the city name stays the same for more than one salesman, then the city name is only printed whenever it changes. This is achieved by comparing each city READ to the previous city READ (before the first READ, the "previous" city would be blanks). Also notice that the cents have been dropped off the dollars, and the result printed in a left-justified column (lined up on left margin). Make up your own set of DATA cards as shown below, being sure to mark the statement numbers into columns (2),3,4 of the optic mark cards, in preparation for question #2. Use "$ Variables" to READ and PRINT the city & salesman columns of names.

City	Salesman	Dollars
RAINY	JONES	2547.60
RAINY	SMITH	9230.25
SNOWY	CLARK	1752.40
WINDY	BROWN	15394.18
WINDY	ADAMS	815.70
WINDY	BLACK	3428.67

2) Code a program to list salesmen, from best to worst in Canada, as shown below:

SALESMAN	COMMISSION	CITY
XXXXX	XXXX.XX	XXXXX
XXXXX	XXX.XX	XXXXX
XXXXX	XXX.XX	XXXXX
XXXXX	XXX.XX	XXXXX
XXXXX	XXX.XX	XXXXX
XXXXX	XX.XX	XXXXX

TOTAL XXXX.XX

The commission received by the salesman is 12 % of the dollars he sold. Notice that the commission is printed to the nearest cent, and is right justified (lined up on right margin). The city is always printed. Use the same DATA cards as for question #1 above, but first sort them into descending order by dollars sold, and mark 1,2,3,... 6 into column 1 of optic mark DATA cards.

3) A general insurance company sells mainly life insurance, but also some car and some house insurance. It has a magnetic disk file containing one record for each life insurance customer who is head of his household. From time to time, the management of the insurance company obtains Exception reports so as to help them operate their business more efficiently. Unlike the Comprehensive reports of questions # 1, 2 above which listed every record on file, an Exception report only lists those records which qualify according to certain criteria. Code a program which skips to the top of a new page, prints the current date, and prints the following exception report(all data is numeric):

TODAY'S DATE

FAMILY-DEAL PROSPECTS

SALESMAN	CUSTOMER	AGE	SALARY	INSURANCE
XXX	XXXXX	XX	XXXXX	XXXXX
XXX	XXXXX	XX	XXXXX	XXXX
XXX	XXXXX	XX	XXXX	XXXXX
XXX	XXXXX	XX	XXXXX	XXXX

Each line in the report is only printed if ALL the following qualifications are met by the customer's record:

> Age is under 30, wife & children not insured, salary is more than $7500, present insurance is less than twice his salary, and health rating is tops.

The disk file is # 1, and is provided by your teacher (password protected). Each disk record contains the following fields (all data is numeric):

Customer #,	Health,	Age,	Marriage?,	Children?,	$Salary,	Insured$,	Salesman#,	Car,	House.

Some of the fields contain a code number, to indicate more details, as follows:

HEALTH: -1 poor.
Ø average.
1 tops.

MARRIAGE: -1 wife insured.
Ø single.
1 wife not insured.

CHILDREN: -# of insured.
Ø no kids.
+# of not insured.

CAR/HOUSE: -1 insured.
Ø do not own.
1 not insured.

4) Code a program to print an exception report with the same format and titles as in problem # 3 above, but with the main title of WELL INSURED, POOR PROSPECTS. The same disk file # 1 is used as for problem # 3 above. However, each line in this report is only printed if both the wife and children are insured, or if the customer is both older than 55 and his insurance is more than 3 times as much as his salary. Also print a line if a customer has all the following characteristics: not married, no children, age over 40 years.

5) Code a program to sequentially read file # 1 above and sequentially write each record into file # 4 while printing one line per record, close spaced. Save file # 4. Code another program which reads one Customer # at a time from a DATA card, and for each Customer # it prints the following two line report, with a blank line before and after the report:

CUST.#	HEALTH	AGE	WIFE	KIDS	SALARY	INSURED	S-M	CAR	HOUSE
XXXXX	X	XX	X	X	XXXXX	XXXXX	XXX	X	X

Then have your program read a DATA card, add the amount read onto the dollars insured, and re-write the record onto file # 4. Check that it works okay by again requesting the above report via a data card for same customer #. Do not check for errors such as "no such customer # ".

6) If you have the opportunity to try running your program from problem # 5 in the "hands-on" mode, replace the READ with INPUT and remove the DATA cards. Note that use of INPUT during batch runs will automatically cause your program to cancel before completing its run.

Also modify your problem # 5 program so that instead of using a READ or INPUT to obtain the amount to be added onto the dollars insured field, two numbers will be READ or INPUT. The first number will be the field number to be replaced (2,3,...,10) and the second number will be the new value which replaces the old value of the field. The program will then re-write the record onto file #4. If the field number supplied is 1, then no field is replaced. Thus, you will now have a "generalized" update program. Subscripted variables is easiest way.

7) Your teacher has a copy of a program which is a simple version of the type of software known as a Generalized Report Generator. This kind of program is usually part of a Management Information System and thus is supposed to be easy to use. Managers are not usually skilled computer programmers. Our simple program uses file #1 from problem # 3 above. It can be used to produce either comprehensive or exception reports of up to 10 columns of numbers (there are 10 fields in our disk file records). For any kind of report, each field of the disk file can be tested or not, as desired. Each tested field is printed as one column of numbers, in the report. If a field is tested twice (or more), it appears in two different columns of the report (remember, we said our program was not sophisticated). For example, a double test is: choose all customers whose age is more than 30 and less than 55 years. To specify which fields are to be tested, code a DATA card using statements # 301, 302, etc. Three numbers are required:

first, the field # (1 to 10) of the field to be tested.

second, the test code #: -1 for LESS THAN.
Ø for EQUALS.
1 for GREATER THAN.

third, the limit # (eg; less than what ?, or equals what ?, etc.)

To signal end-of-data, the last three numbers in the DATA (last card only) must be Ø,Ø,Ø . If you want more than one kind of exception report, your card deck will consist of DATA card(s), RUN card; DATA card(s), RUN card; DATA card(s), RUN card; etc. To produce a title and column headings, use REMARK statements and LIST 4ØØ, as shown below in the example. Only the fields tested will be printed on the report. Which column of the report contains a particular field depends on the order in which the fields occur in the DATA card(s), and not on the position of the field within the disk file records. The report starts printing on the sixth space to the right of space #1.

EXAMPLE:

Print an exception report of all salesman #'s whose customer age is older than 60 years. The salesman# is field #8 in disk record, & always less than 999. Age is field # 3, in disk record.

SOLUTION:

```
3Ø1 DATA 8,-1,999,3,1,6Ø,Ø,Ø,Ø
4Ø1 REM     TOUGH SALES
4Ø2 REM S-M  C-AGE
LIST 4ØØ
RUN
```

PRINTED RESULTS:

```
LIST 4ØØ
4Ø1 REM     TOUGH SALES
4Ø2 REM S-M  C-AGE
RUN
   XXX    XX
   XXX    XX
   XXX    XX
   XXX    XX
```

PROBLEM: Use the above described program to generate an exception report with the title of P R O S P E C T S and the columns 2,3,4 headings of PAY CAR HOUSE (no heading on first column). Print all customers numbers less than 99999 if salary is more than 8572 and if car and house are not insured. Print the customer #'s in first column.

PROGRAMMING PROBLEMS — UNIT # 7

1) A scientific golf player wants to know what angle to hit the ball, to cause it to fly the farthest distance before landing. He is playing on level ground. To solve his dilemna, code a program which will simulate the flight of the ball. Use an asterisk * to show the position of the ball every 30 feet horizontally. Show the position of the ground by printing the letter I at the beginning of each line, down the left margin of the page. After the computer has finished, you will have to rotate the paper one quarter turn, to make the ground level. Print three separate flights, on three separate pages: where the angle of hit is 30, 45, and 60 degrees, respectively. Assume he always hits the ball with the same initial speed of 143 ft/sec.(100 miles/hr), direction of travel. Your printed page will appear approximately as shown here: However, for your printout, only print the asterisk every second line, while printing the letter I on every line. Thus, the distance from line to line represents 20 feet. The number of columns H to space over from the letter I to the asterisk (to the right) can be calculated as follows; where D is 30,60,90, ... 1200 .

$H = ((UT)-(16T^2)) / 9$ where U = Speed times SIN(angle/57.3)

and T = D / X

where X = Speed times COS(angle/57.3)

Note: "times" means multiply. Hint: Use TAB, and stop when H < 0 , after hit.

2) A mathematical golf player decides to solve the same situation as problem # 1 above. However, he derives a mathematical model which predicts the horizontal distance. Code a program which will print two columns of numbers: the first column will be the angle of hit in degrees (30,31,32,...,60); the second column will be the calculated horizontal distance travelled by the ball in air (yards). Also print the column headings: ANGLE DISTANCE . The mathematical model in this case is the following equation; where D is the horizontal distance:

D = G times T/48 , and G = Speed times COS(angle/57.3)

and T = Speed times SIN(angle/57.3)

PROGRAMMING PROBLEMS — UNIT # 9

1) A chemical manufacturing company produces a certain chemical (imaginary) inside of a large, closed container. A computer is used to control the process. It tries to keep the pressure P inside the container at a value of 144 and the temperature T at a value of 156, for a variable time duration. Code a program to control the chemical process. You must use the letters T,P,V,R & this method:

-Computer reads T and P, by using a teacher supplied subroutine. The value of T and P is obtained by executing the statement ... GOSUB 9000 and the student must include the teacher supplied coding in his own program.
-If T is less than or greater than 156, then set V to 1 or -1; else set V to 0 . The value of V causes the heating fuel to be increased, decreased, or unchanged.
-If P is less than or greater than 144, then set R to 1 or -1; else set R to 0 . The value of R turns on a pump,or else a pressure release valve, or nothing.
-Print one line showing the value read for T & P, the resulting V & R, sum T/P .
-Repeat indefinitely, until sum of all T/P exceeds 43.33, then print DRAINED OUT.
-The first statement in your program must be: 1 LET D = 1 .

ANSWERS TO PROBLEMS

UNIT # 1

1)

```
SCRATCH   TEACHER NAME
1Ø LET R = 47.35
2Ø LET A = 3.14159 * R↑2
3Ø PRINT A
9999 END
LIST
RUN     STUDENT NAME
```

2)

```
SCRATCH   TEACHER NAME
1Ø LET L = 8
2Ø LET W = 3
3Ø LET H = 5
4Ø LET V = L*W*H
5Ø PRINT V
9999 END
LIST
RUN     STUDENT NAME
```

3)

```
SCRATCH   TEACHER NAME
1Ø PRINT "AREA","CIRCLE P.",
                "SQUARE P."
2Ø FOR A = 1 TO 8
3Ø LET P = 2 * SQR(A * 3.14159)
4Ø LET Q = 4 * A↑.5
5Ø PRINT A,P,Q
6Ø NEXT A
9999 END
LIST
RUN     STUDENT NAME
```

4)

```
SCRATCH   TEACHER NAME
1Ø PRINT "RADIUS","SURFACE"
2Ø FOR R = 2 TO 17 STEP 3
3Ø LET A = 4 * 3.14159 * R↑2
4Ø PRINT R,A
5Ø NEXT R
9999 END
LIST
RUN     STUDENT NAME
```

5)

```
SCRATCH   TEACHER NAME
1Ø FOR M = .5 TO 1.5 STEP .5
2Ø FOR X = 1 TO 1Ø
3Ø LET Y = M * X + 3
4Ø PRINT TAB (Y) "*"
5Ø NEXT X
6Ø NEXT M
9999 END
LIST
RUN     STUDENT NAME
```

6)

```
SCRATCH   TEACHER NAME
1Ø FOR B = 2 TO 18 STEP 8
2Ø FOR X = 1 TO 1Ø
3Ø LET Y = 2 * X + B
4Ø PRINT TAB (Y) "*"
5Ø NEXT X
6Ø NEXT B
9999 END
LIST
RUN     STUDENT NAME
```

7)

```
SCRATCH   TEACHER NAME
1Ø LET B = 1ØØØ
2Ø LET R = 7
3Ø PRINT B;
4Ø LET I = B * R / 1ØØ
5Ø LET B = B + I
6Ø IF B <= 2ØØØ THEN 3Ø
7Ø PRINT "DONE"
9999 END
LIST
RUN     STUDENT NAME
```

8)

```
SCRATCH   TEACHER NAME
1Ø LET Q = 5
2Ø PRINT Q;
3Ø LET Q = Q / 2
4Ø IF Q>= .1 THEN 2Ø
5Ø END
LIST
RUN     STUDENT NAME
```

9)

```
SCRATCH   TEACHER NAME
1Ø LET X = 2
2Ø PRINT X;
3Ø LET X = 3*X
4Ø IF X<= 1ØØØ THEN 2Ø
999 END
LIST
RUN     STUDENT NAME
```

ANSWERS TO PROBLEMS **UNIT # 3**

1)

```
SCRATCH   TEACHER NAME
1Ø CALL (48)
2Ø READ X,A,B,C,D,Y
3Ø CALL (6,4,3,X,A,B,C,D,Y)
4Ø GO TO 2Ø
5Ø DATA 327,1Ø,1,3,11,15
6Ø DATA 254,1Ø,9,12,12,13
7Ø DATA 436,1Ø,15,8,14,14
8Ø DATA 999,5,14,4,19,Ø
9999 END
LIST
RUN    STUDENT NAME
```

```
SCRATCH   TEACHER NAME
1Ø LET Q = 9E37
2Ø CALL (6,4,1,X,A,B,C,D,Y)
3Ø PRINT X;Q,A;B;C;D;Q,Y
4Ø GO TO 2Ø
9999 END
LIST
RUN    STUDENT NAME
```

4)

```
SCRATCH   TEACHER NAME
5 LET Q = 9E37
1Ø READ E,R,V
2Ø CALL (9,1,2,N,I,S,U,R,N,M,D,H)
3Ø LET T = R + 1.5*V
4Ø LET C = H*T - D
5Ø LET P = INT((C+.005)*1ØØ)/1ØØ
6Ø FOR K = 1 TO 5Ø
7Ø PRINT "*";
8Ø NEXT K
9Ø PRINT TAB(7)"THE WHAT CO. LTD."
1ØØ PRINT TAB(3Ø)"SEPT. 24/72."
11Ø PRINT
12Ø PRINT TAB(14)"PAY $";P
13Ø PRINT
14Ø PRINT TAB(14)"TO  ";Q;I;Ø;S;U;R;N;M
15Ø PRINT
16Ø PRINT TAB(7)"THE UNKNOWN BANK"
17Ø PRINT TAB(7)"WHICH CITY, B.C."
18Ø FOR K = 1 TO 5Ø
19Ø PRINT "*";
2ØØ NEXT K
21Ø GO TO 1Ø
22Ø DATA 1436,4Ø,3,3257,4Ø,6
23Ø DATA 2894,4Ø,Ø
9999 END
LIST
RUN   STUDENT NAME
```

2)

```
SCRATCH   TEACHER NAME
1Ø CALL(48)
2Ø READ X,A,B,C,D,Y
3Ø LET S = X
4Ø CALL (6,4,4,X,A,B,C,D,Y)
5Ø IF X # THEN 2Ø
6Ø CALL (6,4,3,S,A,B,C,D,Y)
7Ø GO TO 2Ø
8Ø DATA 254,1Ø,9,12,12,12
9Ø DATA 173,2,9,12,12,13
1ØØ DATA 436,1Ø,15,1,14,14
9999 END
LIST
RUN    STUDENT NAME
```

3)-Add the following statement to the first program of question # 1 :

```
35 IF X = 999 THEN 9999
```

-Add to the second program of question 1:

```
25 IF X = 999 THEN 9999
```

-Add following to program of question 2:

```
25 IF X = 999 THEN 9999
11Ø DATA 999,1,2,3,4,Ø
```

5) Add the following statements to the program solution for question # 4:

```
2Ø CALL (9,1,1,N,I,S,U,R,N,M,D,H)
22 IF N < E THEN 2Ø
24 IF N = E THEN 3Ø
26 PRINT "NO SUCH EMPLOYEE # AS";E
27 READ E,R,V
28 GO TO 22

22Ø DATA 1436,4Ø,3,2894,4Ø,Ø
23Ø DATA 3Ø25,21,Ø,3257,4Ø,6
```

ANSWERS TO PROBLEMS **UNIT # 5**

1)

```
SCRATCH   TEACHER/STUDENT NAMES
1Ø LET $P = "     "
2Ø PRINT "CITY","SALESMAN",DOLLARS"
3Ø PRINT TAB(31)"SOLD"
4Ø PRINT
5Ø READ $C,$M,D
6Ø LET T = (D + 5Ø) / 1ØØ
7Ø IF $C # $P THEN 1ØØ
8Ø LET $L = "     "
9Ø GO TO 12Ø
1ØØ LET $L = $C
11Ø LET $P = $C
12Ø PRINT $L,$M,T
13Ø GO TO 5Ø
14Ø DATA "RAINY","JONES",25476Ø
15Ø DATA "RAINY","SMITH",923Ø25
16Ø DATA "SNOWY","CLARK",17524Ø
17Ø DATA "WINDY","BROWN",1539418
18Ø DATA "WINDY","ADAMS",8157Ø
19Ø DATA "WINDY","BLACK",342867
9999 END
LIST
RUN  1ØØ
```

2)

```
SCRATCH   TEACHER/STUDENT NAMES
1Ø LET T = Ø
2Ø PRINT "SALESMAN","COMMISSION","CITY"
3Ø PRINT
4Ø FRMT (7,2)
5Ø READ $C,$M,D
6Ø IF $C = "EOF  " THEN 11Ø
7Ø LET P = ( ( D * 12 ) + 5Ø ) / 1ØØ
8Ø LET T = T + P
9Ø PRINT $M,P;F4Ø,$C
1ØØ GO TO 5Ø
11Ø PRINT
12Ø PRINT TAB(6)"TOTAL";T;F4Ø
117Ø DATA "WINDY","BROWN",1539418
215Ø DATA "RAINY","SMITH",923Ø25
319Ø DATA "WINDY","BLACK",342867
414Ø DATA "RAINY","JONES",25476Ø
516Ø DATA "SNOWY","CLARK",17524Ø
618Ø DATA "WINDY","ADAMS",8157Ø
9998 DATA "EOF  ","     ",Ø
9999 END
LIST
RUN  1ØØ
```

3)

```
SCRATCH   TEACHER NAME
1Ø CALL (11)
2Ø CALL (17)
3Ø PRINT TAB(25)"FAMILY-DEAL PROSPECTS"
4Ø PRINT "SALESMAN","CUSTOMER","AGE",
                  "SALARY","INSURANCE"
5Ø PRINT
6Ø CALL (1Ø,1,1,C,H,A,M,K,P,I,R,V,D)
7Ø IF A < 3Ø AND M-1 AND K > Ø AND
      P > 75ØØ AND I < 2*P  AND H-1
                                THEN 9Ø
8Ø GO TO 6Ø
9Ø PRINT R,C,A,P,I
1ØØ GO TO 6Ø
9999 END
LIST
RUN    STUDENT NAME
```

4)

Replace statements # 3Ø and 7Ø of problem # 3 answer with the following:

```
3Ø PRINT TAB(22)"WELL INSURED,
                    POOR PROSPECTS"

7Ø IF M < Ø AND K < Ø OR A > 55 AND
                  I >  3*P  THEN 9Ø
```

Also insert the following statement:

```
75 IF M-Ø AND K-Ø AND A > 4Ø THEN 9Ø
```

5) Continued:

```
SCRATCH   TEACHER NAME
1Ø READ X
2Ø CALL (1Ø,4,2,A,B,C,D,E,F,G,H,I,J)
3Ø PRINT
4Ø PRINT "CUST.# HEALTH AGE etc. HOUSE"
5Ø PRINT A;B;C;D;E;F;G;H;I;J
6Ø PRINT
7Ø READ Q
8Ø LET G = G + Q
9Ø CALL (1Ø,4,4,A,B,C,D,E,F,G,H,I,J)
1ØØ GO TO 1Ø
11Ø DATA 1234,-5ØØØ,1234,Ø
9999 END
LIST
RUN    STUDENT NAME
```

6) For the first part, replace statements 1Ø and 7Ø and remove statement 11Ø from problem # 5 program # 2 as follows:

```
1Ø INPUT X
7Ø INPUT Q
```

For the second part (generalization), the following replacement statements would be required for the problem # 5 program # 2 modification:

```
1Ø INPUT X(1)
2Ø CALL (1Ø,4,2,X(1),X(2),X(3),X(4),X(5),X(6),X(7),X(8),X(9),X(1Ø))
5Ø PRINT X(1);X(2);X(3);X(4);X(5);X(6);X(7);X(8);X(9);X(1Ø)
7Ø INPUT N,Q
8Ø LET X(N) = Q
9Ø CALL (1Ø,4,4,X(1),X(2),X(3),X(4),X(5),X(6),X(7),X(8),X(9),X(1Ø))
11Ø DATA  no longer needed, remove.  Put in   75 IF N = 1 THEN 1Ø
```

7)

```
3Ø1 DATA 1,-1,99999,6,1,8572,9,Ø,1,1Ø,Ø,1,Ø,Ø,Ø
4Ø1 REM    P R O S P E C T S
4Ø2 REM    PAY   CAR   HOUSE
LIST 4ØØ
RUN
```

ANSWERS TO PROBLEMS

UNIT # 7

1)

```
SCRATCH   TEACHER NAME
1Ø LET V = 143
2Ø FOR A = 3Ø TO 6Ø STEP 15
3Ø CALL (11)
4Ø PRINT "I*"
5Ø FOR D = 3Ø TO 12ØØ STEP 3Ø
6Ø LET X = V * COS(A/57.3)
7Ø LET T = D / X
8Ø LET U = V * SIN(A/57.3)
9Ø LET H = ((U*T - 16*T  2)/9) + .5
1ØØ IF H<Ø THEN 14Ø
11Ø PRINT "I"
12Ø PRINT "I";TAB(H)"*"
13Ø NEXT D
14Ø NEXT A
9999 END
LIST
RUN    STUDENT NAME
```

2)

```
SCRATCH   TEACHER NAME
1Ø PRINT "ANGLE","DISTANCE"
2Ø LET V = 143
3Ø FOR A = 3Ø TO 6Ø
4Ø LET G = V * COS(A/57.3)
5Ø LET T = V * SIN(A/57.3)
6Ø LET D = G * T / 48
7Ø PRINT A,D
8Ø NEXT A
9Ø END
LIST
RUN    STUDENT NAME
```

ANSWERS TO PROBLEMS **UNIT # 9**

1) Two sample solutions are shown here. The first is very brief, because it uses the math function SGN. The other is more likely the kind of program an average student might code. Note that both solutions have included the teacher supplied subroutine as required (statements # 9000 onward).

```
SCRATCH   TEACHER NAME
1 LET D = 1
2 LET X = Ø
3 PRINT "P","R","T","V","T/P"
4 GOSUB 9ØØØ
5 LET R = SGN(144 - P)
6 LET V = SGN(156 - T)
7 LET X = X + T/P
8 PRINT P,R,T,V,X
9 IF X <= 43.33 THEN 4
1Ø PRINT "DRAINED OUT"
11 STOP

9ØØØ REM ...'MEASURES' T, P
9Ø1Ø IF D = Ø THEN 9Ø5Ø
9Ø2Ø LET R = V = D = Ø
9Ø3Ø LET P = 144
9Ø4Ø LET T = 156
9Ø5Ø IF INT(P) # 144 THEN 9Ø7Ø
9Ø6Ø LET Z1 = SGN(5 - INT(1Ø * RND(Ø)))
9Ø7Ø IF INT(T) # 156 THEN 9Ø9Ø
9Ø8Ø LET Z2 = SGN(5 - INT(1Ø * RND(Ø)))
9Ø9Ø LET P = INT(P + R - Z1 * INT(2 * RND(Ø)))
91ØØ LET T = INT(T + V - Z2 * INT(2 * RND(Ø)))
911Ø RETURN
9999 END
LIST
RUN    STUDENT NAME
```

```
SCRATCH   TEACHER NAME
1 LET D = 1
2 LET X = Ø
3 PRINT "P","R","T","V","T/P"
4 GOSUB 9ØØØ
1Ø IF P < 144 THEN 5Ø
2Ø IF P > 144 THEN 7Ø
3Ø LET R = Ø
4Ø GO TO 8Ø
5Ø LET R = 1
6Ø GO TO 8Ø
7Ø LET R = -1
8Ø IF T < 156 THEN 12Ø
9Ø IF T > 156 THEN 14Ø
1ØØ LET V = Ø
11Ø GO TO 15Ø
12Ø LET V = 1
13Ø GO TO 15Ø
14Ø LET V = -1
15Ø LET X = X + T/P
16Ø PRINT P,R,T,V,X
17Ø IF X > 43.33 THEN 19Ø
18Ø GO TO 4
19Ø PRINT "DRAINED OUT"
2ØØ END

9ØØØ REM ...'MEASURES' T, P
9Ø1Ø IF D = Ø THEN 9Ø5Ø
9Ø2Ø LET R = V = D = Ø
9Ø3Ø LET P = 144
9Ø4Ø LET T = 156

   etc.
   etc.
   etc.

911Ø RETURN
9999 END
LIST
RUN    STUDENT NAME
```

Notice that in the first sample solution, statements # 5, 6 do the same thing as the second sample solution's statements # 1Ø through 14Ø, which is seven times as many lines. Also notice that one statement can be saved depending upon how you code your final IF ...THEN statement (with 'greater than' or with 'less than or equals'). And you can see, more than one END statement can be used in one program. The report printed by the programs is needed so that humans can monitor the operation of the chemical process while the computer is doing it, from moment to moment. The report also supplies a written history in case the chemical process malfunctions. Then the report aids the debugging.

_____ _____ _____ _____, 1973.
(Teacher) (School) (Course) (Date)

_____
(Enrolment)

Questionnaire for Teachers about Computer-Based Instruction

(Note: Please complete one return for each computer course taught.)

1. What is the ultimate goal of this course? _____

_____

2. List the topics covered and the specific objectives in each one.

Topics	Objectives

3. Indicate the approximate number of students who have been using the following facilities during the school year (September, 1972 to January, 1973). Under 'Frequency of Use' specify how often the facility is used (e.g., every lesson, about once a month, etc.).

Facility	No. of Students	Frequency of Use
Pickup/delivery for Batch Mode BASIC on Hewlett Packard computers		
Hands-on batch mode BASIC in evenings: a) Optic card student programs b) Paper tape student programs		

Facility	No. of Students	Frequency of Use
Hands-on interactive coding of BASIC via portable terminal		
Demonstration and other supplied BASIC programs via portable terminal		
* FORTRAN (or WATFIV) at: a) UBC b) SFU c) Langara d) BCIT e) Other _____ _____ _____		

* Specify whether optic mark or keypunching was used for items (a) through (e).

For 'Other', specify other languages and hardware, if any.

4(a) List the reference materials used by your students, the number of students using them, and the frequency of use.

Reference Materials	No. of Students	Frequency of Use
Catalogue items (specify):		
_____	_____	_____
_____	_____	_____
_____	_____	_____
Books (specify):		
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
Other materials (specify):		
_____	_____	_____
_____	_____	_____

4(b). List the reference materials used by yourself and the frequency of use.

Reference Materials	Frequency of Use
Catalogue items (specify):	
_____	_____
_____	_____
_____	_____
Books (specify):	
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
Other Materials (specify):	
_____	_____
_____	_____

5. What percent of your teaching time do you spend instructing computer courses?

_____%

6. What teaching certification do you have? _____

7. What degrees do you hold? _____

8. List any courses you have taken in computing science _____

_____

9. What background experience have you had with computers?

_____

_____

10. Complete the following table for the students in your course.

	Number of Years of Computer Experience						
	0	1	2	3	4	5 or more	Total
Grade 8							
Grade 9							
Grade 10							
Grade 11							
Grade 12							
Other (e.g. Occup.)							
Total							

11. List any general characteristics that apply to the students in your computer course (e.g., motivation, scholastic aptitude). _____

_____

12. What percent of your teaching time do you spend working with:

a) individuals _____%
b) small groups _____%
c) the entire class _____%

Total = 100 %

13. What is the frequency of instruction in this course?

_____periods (of _____ minutes each) in a cycle of _____days.

14. What procedures do you use to evaluate student achievement? _____

_____

_____

_____

15. In which of the following subject areas are the problems on which your students are working? (Rank in order of frequency only those that apply.)

En_____ SS_____ Gu_____ Hi_____ Law_____ Ec_____ Ma_____

Sc_____ Bio_____ Ch_____ Ph_____ Co_____ Mu_____ Games_____

Surveys_____ Others (specify)_____

16. How many times during the year have the following class activities been provided? Approximately how many students have participated? Include any additional enrichment activities not listed.

Activity	Number of Times	Number of Students
(a) audio-visual presentation		
(b) newspaper clippings		
(c) magazine articles		
(d) research assignments		
(e) field trips (specify locations) _____		
(f) _____		
(g) _____		

17. List what you consider to be the strengths and weaknesses of the computer-based instructional program in your school and make any suggestions you might have for its improvement and extension.

(a) Strengths	(b) Weaknesses

(c) Suggestions for improvement and extension

_____

_____

_____

18. To indicate the range of difficulty of the topics on which your students are working, kindly include a sample printout of an easy, an average, and a difficult problem.

Grade_____ School_____ Computer Course_____ Date_____

Student Questionnaire Regarding Computer-Based Instruction

1. In respect to difficulty, how did you find the computer course? (Check one)

   _____ (A) easier than I had expected

   _____ (B) harder than I had expected

   _____ (C) about the same as I had expected

2. The work load involved in the computer course was: (Check one)

   _____ (A) heavier than I had anticipated.

   _____ (B) lighter than I had anticipated.

   _____ (C) about the same as I had anticipated.

3. The course itself was: (check one)

   _____ (A) highly interesting.

   _____ (B) fairly interesting.

   _____ (C) not interesting.

4. How many hours per week did you spend (outside of class time)

   _____ (A) on required work for the computer course?

   _____ (B) on computer work for your own interest and/or enjoyment?

5. On the average, did you spend more or less time outside of class on the computer course than on other courses? (Check one)

   _____ (A) more

   _____ (B) less

   _____ (C) about the same

6. If you spent more time, what accounted for the extra time?

_____ (A) heavy work load

_____ (B) interest in the course

_____ (C) other (specify) _____

7. Have you had sufficient opportunities to get help from your teacher? (Check one and comment if necessary)

_____ (A) Yes

_____ (B) No Comment_____

8. Rank the following according to their importance as sources of help in the computer course. (i. e., Insert "1" for the source that helped you the most, "2" for the second most important source of help, etc.) Any additional sources of help you consider important but are not listed, may be entered beside "Other" and ranked accordingly.

_____ other students _____ teachers

_____ reference material _____ other_____

_____ computer error messages

9. Did you write programs for other courses

(A) as assignments from teachers ? (Check one)

_____ Yes Name the courses_____

_____ No

(B) on your own initiative? (Check one)

_____ Yes Name the courses_____

_____ No

Comment, if necessary_____

10. Have the thinking processes that you have developed in the computer course helped you in any other courses?

_____

What courses? _____

How? _____

11. Please list what you consider to be the strengths and/or weaknesses of the computer course. In addition list any suggestions you might have to improve the course.

Strengths

1. _____
2. _____
3. _____
4. _____
5. _____

Weaknesses

1. _____
2. _____
3. _____
4. _____
5. _____

Suggestions for improvement

1. _____
2. _____
3. _____
4. _____
5. _____

12. The use of computers is increasing.

    (a) How is this good for people?

    _____

    _____

    (b) How is this bad for people?

    _____

    _____

13. How will this course help you in the future?

    _____

    _____

14. Do you plan to pursue a career in the computer field?

    _____ (A) Yes

    _____ (B) No

    _____ (C) Undecided

15. On your last report card, what was your letter grade standing in the computer course?

    _____

COMPUTER SCIENCE

Date: _____ Name: _____

School: _____

Grade: _____

INSTRUCTIONS:

1. You have one class period to answer ONE of the following problems. Be sure that you read over all questions before you make your choice.

2. If your problem does not work on the first run, you may correct it and re-run it as many times as you wish as long as you hand in a printout for every run you make.

3. Your program control cards should be labelled like this:

   SCRATCH your teacher's name..your school

   LIST

   RUN..your name your grade

NOTE: Remember that you are to answer only ONE problem.

RECORD OF COMPUTER RUNS

Run No.	Date	Reason for re-run
1		
2		
3		
4		
5		
6		
7		
8		

PROBLEM 1:

Temperatures can be measured on both the Fahrenheit and Celsius (Centigrade) temperature scales. When Fahrenheit temperatures (F) are given, the equivalent Celsius temperatures (C) can be calculated using this formula:

$$C = \frac{5}{9} \times (F - 32)$$

Mercury is liquid at room temperature. It has a freezing point of -38.87° C and a boiling point of 356.58° C. This means that mercury will be:

a) in a solid state for Celsius temperatures less than -38.87° C.
b) in a liquid state for Celsius temperatures from -38.87° C to 356.58° C.
c) in a gaseous state for Celsius temperatures higher than 356.58° C.

Write a program that will:

a) Print the title TEMPERATURES AND STATES OF MERCURY
b) Print column titles FAHRENHEIT TEMPERATURE CELSIUS TEMPERATURE STATE OF MERCURY
c) Read the Fahrenheit temperature (F) of a mercury sample.
d) Calculate the equivalent Celsius temperature (C) of the mercury sample.
e) Determine whether the sample is in a solid, liquid or gaseous state.
f) Print under appropriate column titles, values for F, C and one of the comments: SOLID, LIQUID or GAS.

Use these values of F as data for your program:

32, -40 , 0, 1.65E6, 911, -2.25, 455, -66, 763, -25, 1111

Include enough additional data to prevent the computer from printing the out of data error (ERROR 56) at the end of your printout.

Some of your printout should look like this:

```
 RUN ..YOUR NAME   CLASS

TEMPERATURES AND STATES OF MERCURY

FAHRENHEIT      CELSIUS         STATE OF
TEMPERATURE     TEMPERATURE     MERCURY

 32              0              LIQUID
-40             -40             SOLID
```

Write out your program listing (including SCRATCH, LIST and RUN commands) on the BASIC coding form provided.

Code your program on cards and submit it for running on the computer.

PROBLEM 2:

The standard pay rate at the ABC Repair Company is $4.30 per hour up to and including 40 hours per week. The overtime pay rate is time and one half the standard rate (1.5 times $4.30) for all hours worked in excess of 40 per week. Gross pay (G) is calculated by adding the standard pay to the overtime pay. A deduction (D) of 20% (use .20) of gross pay is made for taxes. Net pay (N) is calculated by subtracting the deductions from the gross pay.

Write a program that will:

a) Print out the title ABC REPAIR COMPANY PAYROLL
b) Print out the column titles

TIME WORKED (HOURS)	GROSS PAY ($)	DEDUCTIONS ($)	NET PAY ($)

c) Read the number of hours worked per week (H).
d) Reject any incorrect input data by printing out the hours worked followed by the comment INVALID DATA.
e) Calculate values for gross pay (G), deductions (D) and net pay (N) for each value of H.
f) Print under their appropriate column titles, values for H, G, D and N.

Use these values of H as data for your program:

40, -2, 48, 15.5, 0, 41.5, 54, 40.5, -3.5, 44

Include enough additional data to prevent the computer from printing the out of data error (ERROR 56) at the end of your printout.

Some of your printout may look like this:

```
RUN ..YOUR NAME   GRADE

ABC REPAIR COMPANY PAYROLL

TIME WORKED     GROSS PAY       DEDUCTIONS      NET PAY
(HOURS)         ($)             ($)             ($)

 40              172             34.4            137.6
-2              INVALID DATA
```

Write out your program listing (including SCRATCH, LIST and RUN commands) on the BASIC coding form provided.

Code your program on cards and submit it for running on the computer.

PROBLEM 3:

Write a program that will read in a number, X, and then print out values for X, 4-X, $\sqrt{X}$ and 1/X under these column titles:

X 4-X X↑.5 1/X

Include checks for undefined answers. (Hint: These may occur for $\sqrt{X}$ and 1/X). If an undefined answer is found, have the computer print the message UNDEFINED under the appropriate column title.

Write a program that will:

a) Print the main title PROPERTIES OF X.
b) Print the column titles X 4-X X↑.5 1/X
c) Read a value of X.
d) Identify any values of X that will cause one or more of the expressions 4-X, X↑.5 or 1/X to be undefined.
e) Calculate values for 4-X, $\sqrt{X}$ and 1/X.
f) Print, under their appropriate column titles, values for X, 4-X, $\sqrt{X}$ and 1/X or the message UNDEFINED.

Use these values of X as data for your program:

25, -4, 81, -81, 9, -36, 0, 2.25 E6, 1.69 E-8, 64.76, -127.8

Include enough additional data to prevent the computer from printing the out of data error (ERROR 56) at the end of your printout.

Some of your printout may look like this:

```
 RUN ..YOUR NAME   GRADE

PROPERTIES OF X

X              4-X            X↑.5           1/X

 25            -21             5.             4.00000E-02
-4              8              UNDEFINED      -.25
```

Write out your program listing (including SCRATCH, LIST and RUN commands) on the BASIC coding form provided.

Code your program on cards and submit it for running on the computer.

THE COMPUTER PROGRAMMER APTITUDE BATTERY

The Computer Programmer Aptitude Battery (CPAB) was developed by Jean Maier Palormo of Science Research Associates, Inc., to aid data-processing managers and personnel directors in selecting persons with the aptitude for computer programmer and systems analyst positions. It is generally administered to university graduates or high school graduates with technical training who are seeking such positions. The Battery comprises five separately timed tests, measuring the following skills and aptitudes:

Verbal Meaning (38 items - 8 min.)	a test of communications skill; vocabulary commonly used in mathematical, business and systems engineering literature
Reasoning (24 items - 20 min.)	a test of ability to translate ideas and operations from word problems into mathematical notations
Letter Series (26 items - 10 min.)	a test of abstract reasoning ability, finding a pattern in the given series of letters
Number Ability (28 items - 6 min.)	a test of facility in using numbers; ability to estimate quickly reasonable answers to computations
Diagramming (35 items - 35 min.)	a test of ability to analyze a problem and order the steps for solution in a logical sequence.

The CPAB was administered to a Grade 11 computer science class at Templeton Secondary School in Vancouver as a pre-test in January, 1973 (at the beginning of the course), and as a post-test in June, 1973 (upon the completion of the course). Mrs. Zelter, the instructor of the course, did not administer the Diagramming section in January because of a time limitation:

An analysis by "t" test of the mean scores of the computer science class on the pre- and the post-test was made (see Table E-1). Although the post-test performance of the typical student improved on all sections of the Battery, the gains made on the Verbal Meaning, Letter Series and Number Ability subtests were not statistically significant.

To get a general idea of how the students fared on the CPAB, their performance on the battery (the post-test results) was compared to that of two groups for which published norms were available: a group of computer programmer trainees and applicants, and a group of experienced computer programmers

and systems analysts. The educational level of both groups was considerably higher than that of the computer science students.

Two-thirds of the computer programmer trainees and applicants were applying for jobs with a civil service agency in the eastern United States and the remainder were enrolled for introductory computer systems training at universities or computer manufacturer sites. Approximately half of this group were college graduates.

The experienced computer programmers and systems analysts included personnel from a variety of business and industrial installations, including computer manufacturers. Approximately 80% of these were college graduates, and their median experience in the computer programmer field was three to four years.

A comparison of the mean raw scores of the computer science students (after they had finished their semester-length course) with that of the programmer trainees and the experienced programmers on the CPAB is presented in Table E-II.

The performance of the computer science students on the CPAB was impressive. Their average score was better than that of the programmer trainees and applicants on four of the five sub-tests, and better than the average score of the experienced programmers and analysts on three of the tests. The average score on the Total Battery for the grade 11 students far exceeded that of the programmer trainees (85.73 compared with 64.86) and was quite close to the average score of the experienced programmers. The only section of the test on which the computer science students scored lower than both norm groups was Verbal Meaning. They did exceedingly well on the Diagramming section (mean score = 28.13 compared to 25.46 for the experienced programmers), and on the Letter Series section (mean score = 15.70 compared to 11.72 for the experienced programmers).

The fact that the computer science students compared favourably with programmer trainees prior to taking the course (pre-test mean scores for the students were higher than those of programmer trainees on three of four sub-tests) might indicate that students with abilities that would adapt well to a programming career are attracted to the course. It is significant that a one semester course in computer science improved the performance of the students so that they compared favourably to the group of experienced programmers and systems analysts on all but the Verbal Meaning section of the Computer Programmer Test Battery. The statistically significant improvement on the Reasoning section is especially noteworthy.

TABLE E-I: ANALYSIS BY "t" TEST OF PRE- AND POST-TEST SCORES OF COMPUTER SCIENCE STUDENTS (N = 15) ON FOUR TEST SECTIONS OF THE COMPUTER PROGRAMMER APTITUDE BATTERY

	Verbal Meaning		Reasoning		Letter Series		Number Ability	
	Pre	Post	Pre	Post	Pre	Post	Pre	Post
Mean Raw Score	10.53	13.80	11.87	15.00	13.87	15.67	12.73	13.13
Standard Deviation	4.96	4.71	2.72	3.67	3.88	3.32	3.40	4.03
Difference Between Means	3.27		3.13		1.80		0.40	
"t" Value	1.79 (n. s. d.)		2.56*		1.32 (n. s. d.)		0.28 (n. s. d.)	

Legend: n. s. d. -- no significant difference
* -- significant at .02 level

TABLE E-II: MEAN RAW SCORES OF THE GRADE 11 COMPUTER SCIENCE CLASS, THE PROGRAMMER TRAINEES, AND THE EXPERIENCED PROGRAMMERS ON THE COMPUTER PROGRAMMER APTITUDE BATTERY

	Computer Science Class (N = 15)	Programmer Trainees (N = 298)	Experienced Programmers (N = 229)
Verbal Meaning	13.80	16.77	21.58
Reasoning	15.00	9.49	14.77
Letter Series	15.70	10.46	11.72
Number Ability	13.13	12.13	15.90
Diagramming	28.13	16.15	25.46
Total Battery	85.73	64.86	89.39

APPENDIX F

A PREVIOUS STUDY OF COMPUTER-BASED INSTRUCTION IN VANCOUVER SECONDARY SCHOOLS

In 1970, "An Evaluation of Student Experience with Computers in the Instructional Program of Two Secondary Schools of Vancouver, 1969-70" was carried out. In that study, the recommendations of the teachers for expanding and improving the computer program were that:

1) computer courses be offered for credit on an elective basis to students in Grades 9 - 12,
2) Grade 8 students be introduced to computer programming in their regular school subjects,
3) terminals be installed in schools which do not have a computer,
4) batch-processing of computer programs be accelerated,
5) student aides be paid for delivering programs for computer processing,
6) afternoon and evening sessions be expanded for teachers and students who want to use the computer,
7) in-service meetings on computer programming be instituted for teachers,
8) a teacher-expert be appointed to coordinate computer instruction in Vancouver schools,
9) all future computer equipment purchased or leased be able to accept the BASIC language, and
10) long-range policy be formulated in regard to the organization of administrative and educational computer facilities.

As of June, 1973, recommendations (1) and (2) had been implemented in five Vancouver schools. Although no permanent terminals have been installed (item 3, above), a portable terminal is being used by schools on a rotational basis. High speed batch-processing computer facilities (items 4, 5 above) now provide twice daily return of programs, and two drivers have been hired to provide a pickup and delivery service. Afternoon and evening sessions (item 6, above) have been expanded to be available to all secondary schools, in-service training for teachers (item 7, above) has been instituted, and Mr. Wayne Dodds has been appointed (item 8, above) as Computer Consultant for the Vancouver School Board. Recommendations (9) and (10) have also been put into effect.

SURVEY OF STUDENT USAGE OF THE HEWLETT-PACKARD COMPUTERS

Mr. Wayne Dodds, Computer Consultant for the Vancouver School Board, annually conducts a survey of student usage of the Hewlett-Packard computers (currently in operation at John Oliver and Point Grey Secondary Schools).

Table G-I shows the number of schools, teachers and students that have used the computer facilities during the past three years. The table includes students who used the computer in regular subject areas as well as those enrolled in special computer courses and members of computer clubs.

The increase from 700 students in 1970/71 to 5100 students in 1972/73 represents more than a 600% increase in the number of students using the computer facilities. Growth is expected to continue.

A breakdown of computer usage according to subject area is presented in Table G-II. The most growth occurred in the areas of mathematics and science. Mr. Dodds noted the need for the improvement of the computers (by purchase of additional hardware) to better suit the requirements of the commerce program.

This year, eight secondary schools involved half or more of their Grade 8 students by having a special unit of study (two to four weeks) on the computer, as part of their regular mathematics courses. Four of the schools used the computer with some of their General Mathematics students. In addition, a total of 220 elementary students from eight schools used the computer facilities during the 1972/73 school year.

The percentage of students using the computer varied considerably among schools--from 4% to 44% (see Table G-III). There was also considerable variation among schools in the amount of time spent using the portable terminal (see Table G-IV). Students in one school used the terminal on a total of 40 days; other schools never requested to use it. In all, the terminal was used on 146 days (out of a possible 200 days) throughout the school year, during which time it was actually in use for a total of 238 hours (an average of 1.63 hours per day).

An analysis of the computer programs run (via batch processing) is automatically produced daily by the computer and summarized twice a month (see Table G-V). The statistics generated enable the computer consultant to predict what demands will be made on the computer system in the future and what need there may be for improved equipment to handle the volume of work."

Demands were made on the computer to run student programs throughout the entire 1972-73 school year (see Figure G-1). The low points on the graph corresponded to the opening of school in September, the semester change in January, the weeks before Easter vacation (when mid-term exams are held in some classes) and the closing of school in June.

TABLE G-I: SUMMARY OF COMPUTER USAGE DURING THE 1970/71, 1971/72 AND 1972/73 SCHOOL YEARS

School Year	Number of Schools	Number of Teachers	Number of Students
September/70-June/71	4	7	700
September/71-June/72	16	66	3400
September/72-June/73	18	77	5100

TABLE G-II: SUMMARY OF COMPUTER USAGE, BY SUBJECT AREA, DURING THE 1971/72 AND 1972/73 SCHOOL YEARS

	No. of Schools (out of 18)				No. of Teachers				No. of Students			
Year	Math	Science	Commerce	Computer	Math	Science	Commerce	Computer	Math	Science	Commerce	Computer
71-72	13	2	13	11	24	2	23	15	1631	119	857	766
72-73	16	6	11	11	36	11	14	14	2571	746	815	844

TABLE G-III: SUMMARY OF COMPUTER USAGE DURING THE 1972/73 SCHOOL YEAR FOR 18 VANCOUVER SECONDARY SCHOOLS (IN DESCENDING ORDER IN TERMS OF PERCENTAGE OF ENROLMENT USING THE COMPUTER)

School Name	Math	Science	Commerce	Computer	Club	Elementary	Total	% of Enrolment
A	2 - 190	2 - 329	1 - 195	1 - 34	1 - 10	1 - 20	778	44
B	4 - 326			1 - 43	1 - 10	1 - 29	379	30
C	4 - 265			1 - 18			283	29
D	2 - 62	1 - 62	2 - 160	3 - 320		1 - 90	652	27
E	5 - 351		1 - 55		1 - 9		415	23
F	2 - 125	4 - 202	2 - 85	1 - 21			433	20
G	2 - 130		1 - 51	1 - 35	1 - 20	1 - 80	316	20
H	2 - 214		2 - 109				323	18
I	3 - 227		1 - 8	1 - 26			261	17
J	2 - 120	2 - 80		1 - 34	1 - 10		244	15
K		1 - 75					75	15
L	1 - 70			1 - 159			229	13
M	2 - 115		1 - 75	2 - 55			245	12
N	1 - 210		1 - 30					
O	2 - 46			1 - 35	1 - 10		91	9
P	1 - 30						30	9
Q		1 - 40	1 - 25		1 - 15		80	4
R	1 - 30		1 - 22				52	4
Totals	36 -2511	11 - 746	14 - 815	14 - 780	7 - 84	4 -219	5155	18

TABLE G-IV: SUMMARY OF USAGE OF PORTABLE TERMINAL DURING THE 1972/73 SCHOOL YEAR

School	Total Number of Days on which Terminal was Used	Total Number of Hours of Terminal Use
A	40	60.50
B	1	1.00
C	0	0.00
D	1	1.50
E	11	20.25
F	10	15.00
G	17	22.00
H	0	0.00
I	6	8.25
J	12	19.50
K	0	0.00
L	10	17.75
M	6	9.00
N	0	0.00
O	11	26.50
P	5	9.00
Q	16	27.25
R	0	0.00
Total	146	237.50

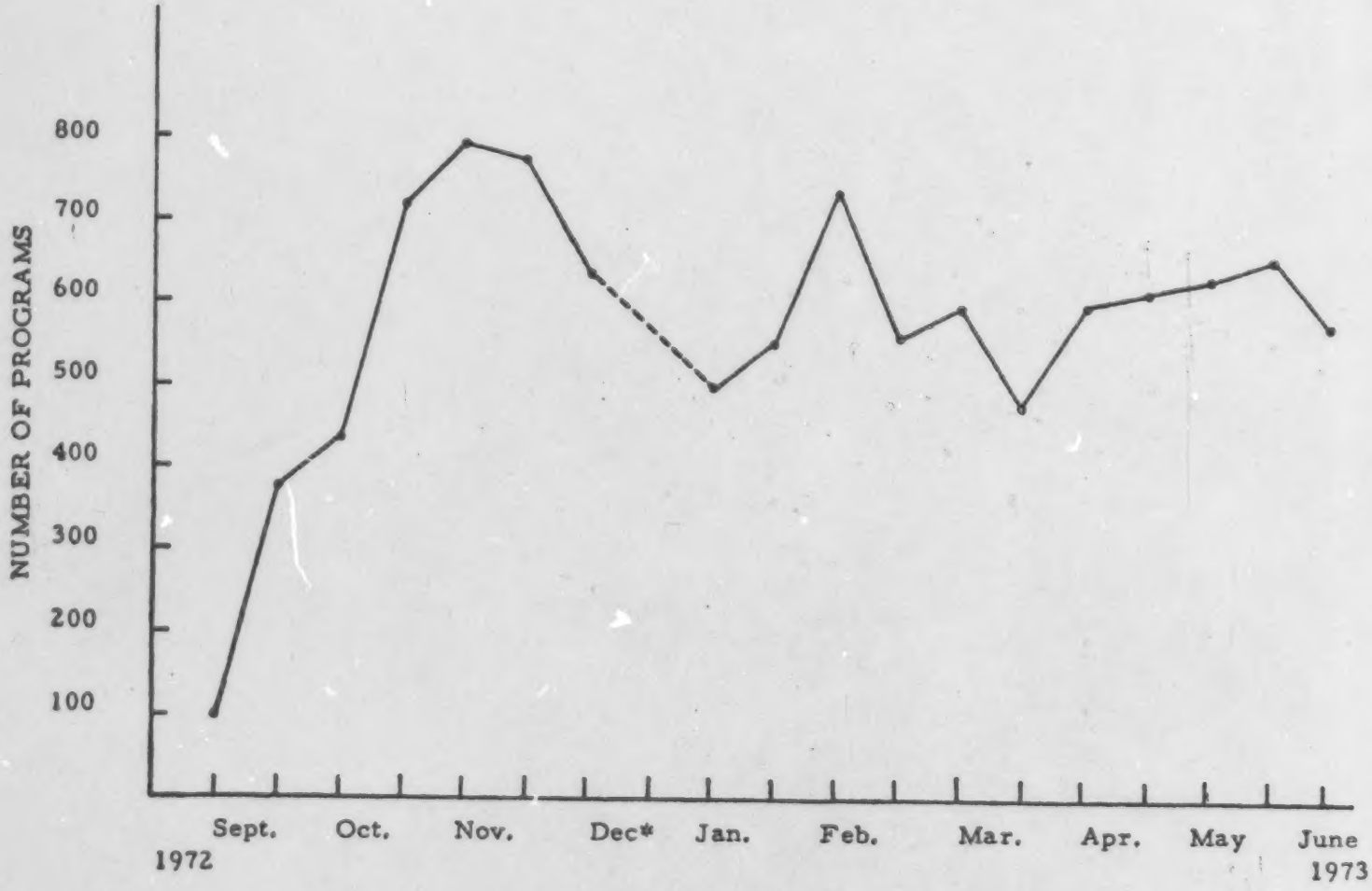
TABLE G-V: SAMPLE OF THE SUMMARY OF PROGRAMS BATCHED, PRODUCED TWICE MONTHLY BY THE COMPUTER

NOV 17/72 AM BATCH ANALYSIS, MONTHLY STATISTICS

	AVERAGE PER DAY	MAXIMUM PER DAY	YEAR-TO-DATE TOTALS	AVERAGE PER PROGRAM
PROGRAMS RUN	443	748	8323	---
ERROR BAIS*	11	16	181	2 %
TIME ,COMPUTER	25	38	9	3
CARDS READ	6912	11036	128597.	16
PAGES USED	588	915	11166	1.3
TIME,ELAPSED	129	190	44	17
...TIMES IN...	MINUTES	MINUTES	HOURS	SECONDS

*Time limit exceeded--execution cancelled automatically.

FIGURE G-1: AVERAGE NUMBER OF PROGRAMS RUN DAILY, 1972-1973 CALCULATED SEMI-MONTHLY



*The computer was not in operation during the Christmas holidays.

APPENDIX H

Tic Tac Toe Game

```
RUN

T I C   T A C   T O E   G A M E .

PURPOSE:  GET 3 X'S IN A LINE BEFORE COMPUTER GETS 3 O'S .
EACH TIME COMPUTER PRINTS ? MARK, YOU MUST TYPE YOUR ANSWER.

TYPE A NUMBER BETWEEN 1 AND 500.
?36
TYPE NUMBERS 1TO 9 TO ANSWER WHERE YOU PUT EACH X,BELOW:

1 2 3
4 5 6
7 8 9

YOU ARE X, I AM O
YOU GO FIRST.
      YOUR X?3
- - X
- O -
- - -
      YOUR X?9
- - X
- O O
- - X
      YOUR X?4
- O X
X O O
- - X
      YOUR X?8
- O X
X O O
O X X
      YOUR X?1
STALEMATE.  NOBODY WINS.


WOULD YOU LIKE TO PLAY AGAIN? INPUT 1 IF YES, 0 IF NO?1
TYPE NUMBERS 1TO 9 TO ANSWER WHERE YOU PUT EACH X,BELOW:
```

APPENDIX I

Lunar Module Game

```
RUN
CONTROL CALLING LUNAR MODULE. MANUAL CONTROL IS NECESSARY
YOU MAY RESET FUEL RATE K EACH 10 SECS TO 0 OR ANY VALUE
BETWEEN 8 & 200 LBS/SEC. YOU'VE 16000 LBS FUEL. ESTIMATED
FREE FALL IMPACT TIME-120 SECS. CAPSULE WEIGHT-32500 LBS
FIRST RADAR CHECK COMING UP

COMMENCE LANDING PROCEDURE
TIME,SECS   ALT,MILES+FEET   VELOCITY,MPH      FUEL,LBS   FUEL RATE
     0          120   0            3600          16500       K= ?0
    10          109   5015.98      3636          16500       K= ?0
    20          99    4223.94      3672          16500       K= ?0
    30          89    2903.94      3708          16500       K= ?100
    40          79    2510.69      3544.6        15500       K= ?0
    50          69    3059.19      3580.6        15500       K= ?100
    60          59    4580.44      3410.87       14500       K= ?150
    70          50    4147.72      3125.48       13000       K= ?0
    80          42    283.352      3161.48       13000       K= ?200
    90          33    4468.01      2742.56       11000       K= ?200
   100          26    4482.8       2289.27       9000        K= ?200
   110          21    871.386      1795.99       7000        K= ?200
   120          16    4830.28      1255.62       5000        K= ?200
   130          14    1274.88      658.919       3000        K= ?185
   140          13    1284.79      49.002        1150        K= ?100
   150          13    3046.78     -292.948       150         K= ?0
   160          14    1799.35     -256.948       150         K= ?10
   170          15    309.899     -259.984       50          K= ?200
FUEL OUT AT  170.25     SECS
ON THE MOON AT 437.79     SECS
IMPACT VELOCITY OF 684.452    M.P.H.
FUEL LEFT IS 0    LBS.
SORRY, BUT THERE WERE NO SURVIVORS-YOU BLEW IT! IN FACT
YOU BLASTED A NEW LUNAR CRATER 190.125    FT. DEEP
TRY AGAIN? ANSWER 1 FOR YES, 0 FOR NO. YOUR ANSWER?1
FIRST RADAR CHECK COMING UP
```

 BEST COPY AVAILABLE

CATALOGUE REFERENCES & AIDS FOR HP COMPUTER USERS April 1, 1973

Educational Basic Language (100's)

Item #	Description	Per Teacher	One Per Student
101	CARD IMAGES, HP BASIC, full size, on looseleaf size paper, 6 images to a sheet, 3 to a side. Can be used for beginners to practice, as exam answer sheets, preparation of examples, etc.		
102	ERROR CODES ... EDUCATIONAL BASIC, on looseleaf size paper, printed both sides: Code #'s, with Meanings, and with Probable Causes. Single page.		X
103	EDUCATIONAL BASIC, summary of all Commands, Statements, Operators, Matrices, Card image, and Examples, on single page, both sides		X
104	HOW TO MARK CARDS ... and FUNDAMENTAL USAGE OF BASIC, on single page, looseleaf size paper, both sides. Contains large example of HP card and how to mark coding onto it, together with all the rules of BASIC which refer to cards.		X
105	A GUIDE TO HP EDUCATION BASIC, a 100 page paper back textbook by the computer manufacturer. It has been found to be of little use to the beginning teacher, (even less use to students) but is useful to moderately experienced teachers of computer topics, as a quick memory aid, etc.	1	X
106	HOW TO USE ALPHA DATA, is a single looseleaf size page, printed both sides, with code numbers, general rules, and five example programs (regular, Disk, Matrices, use of IF, and "packing"). Can be used with all Vancouver School Board versions of BASIC.		X
107	HOW TO USE CALL (11), CALL (17), & CHOOSE JOB PRIORITY, is a single looseleaf page, containing general rules and example programs, and also containing the times, in milliseconds, required to do each operation, and the limits, under each priority: A, B, C.		X
108	EXAMPLE USES OF THE MATH FUNCTIONS, is a single looseleaf size page, printed both sides, showing how and why to use BASIC's built-in functions.		X
109	HOW TO USE THE PORTABLE TERMINAL, is a single looseleaf size page , with cautions & procedures.	5	
110	PROGRAM DIRECTORY FOR THE PORTABLE TERMINAL, is a several page list and description & request-code number of programs stored on tape at each computer.	1	
111	HOW TO SAVE CORE MEMORY, TO ALLOW BIGGER PROGRAMS, is a few pages of rules and examples, which assumes familiarity with the coding techniques involved.		X

BEST COPY AVAILABLE

[redacted] (200's)

Item #	Description	per Teacher	One per Student
201	DISK [redacted] single page, looseleaf size paper, both sides. Is a summary of all the CALL's possible, file sizes, error Code #'s, Meanings, and Probable Causes		X
202	DISK FILES FOR ED. BASIC (SIMULATED), is a dozen paged pamphlet textbook by Mr. W. Dodds, VSB. Contains example coding, purposes, and detailed rules of all the features of Disk File Basic software, as developed by Mr. W. Dodds for VSB.	10	
	[redacted] (300's)		
301	HOW TO USE 'BUSINESS' BASIC, is a 5 page summary of all the differences and extras, with rules & examples. Character data, format, card files. (Assumes user already knows regular BASIC).	20	
	Teacher Aids, General (400's)		
401	CODING SHEETS, on single page, looseleaf size. Is an aid to students to organize their work, by coding statement #'s and body under the appropriate columns, title their program, etc. Can easily be run off by each school for own use.	1	
402	MOVIES ON DATA PROCESSING, is a two dozen paged booklet combining the lists of free movies loaned by several companies (4 or 5 month lead time).	1	
403	REFERENCE TEXTS FOR TEACHERS is a looseleaf page list of titles, authors, and publishers; and usefulness in Vancouver's setting is indicated, where possible. Purchase cost of these books is each school's responsibility, unless specially arranged otherwise.	1	
404	HOW TO MARK & SUBMIT DATA & RUN CARDS FOR UTILITY AND SIMULATION PROGRAMS IN BASIC, is a 3 page explanation, with large pictures of the BASIC card, intended for non-programmers, etc.	30	
405	ED. BASIC OVERHEAD TRANSPARENCY OF CARD, greatly enlarged, for use with overhead projectors.	1	
406	TEST & SURVEY CARD OVERHEAD TRANSPARENCY (large) for explaining how to mark multiple-choice card.	1	
407	TEST SCORER PROGRAM, TEACHER'S INSTRUCTIONS, explains how to use the multiple-choice test scoring service, via computer pickup/delivery.	1	

BEST COPY AVAILABLE

Item #	Description		
408	SURVEY ANALYSIS PROGRAM, TEACHER'S INSTRUCTIONS, explains how to use multiple-choice survey analysis service, via computer pickup/delivery.	1	
409	TEACHERS' GUIDE TO CURRICULUM COMPUTER SERVICES, explains how your students can use VSB computers.	1	
[illegible] Course	(#500's)		
Item #	Description		
501	COMPUTER APPLICATIONS, is a course topic outline prepared by Mr. Dodds, as an introductory, survey type of course for credit, aimed at grade 11, but easily modified for other grade levels. It is submitted once yearly to Victoria Dept of Ed for request to offer as a credit course, experiment. There is a 20 page teacher's guide which doubles as student reference, plus a 9 page set of questions plus a 5 page set of answers, plus other reference materials.	5 5 1	1
Commerce Via Computers	(#600's)		
601	COMPUTERS FOR COMMERCE, INTRODUCTORY UNIT, is a 10 lesson, detailed, 21 pages set of lesson plans exercises, solutions, suitable for beginners, by Mr. Dodds.	5	
Math Via Computers	(#700's)		
701	COMPUTERS FOR MATH, is a 10 lesson introduction by Mr. Dodds. A detailed, 19 page set of lesson plans, exercises, solutions. Suitable for beginners, Grades 8 to 12 Math Courses.	1	
Science Via Computers	(800's)		
801	HOW TO USE POINT GRAPH PLOTTER (BATCH RUN VERSION) explains how to use this program which is already written and tested and works. It can plot and do graph analysis on whole class, for each student.	15	
802	SCIENCE 9 HORSEPOWER EXPERIMENT CALCULATIONS, explains how each student can use a program which is already written, to do or check his calcs.	30	
803	SCIENCE 10 UNIT VI, SIMULATION OF EXPERIMENT 3, explains how to use this program which is already written. Also example of how to simulate, in general.	15	
• • • •	MATH, SCIENCE, COMMERCE, SOCIAL STUDIES: -Also phone Mr. Dodds (731-1131 local 260 or 273) regarding many textbooks and other professionally published materials, which are available to you.	1	

BEST COPY AVAILABLE

Examples of Programs Written by Computer Science 11E Students

1]

```
 SCRATCH DEBBIE JOYCE BLOCK 00)
JAN 17/73--A

5  FOR X=1 TO 6
10  PRINT "DEBBIE JOYCE"
15  PRINT "1036 EAST 20TH AVE."
20  PRINT "874-3641"
25  PRINT
30  PRINT
35  NEXT X
9999  END

 RUN
DEBBIE JOYCE
1036 EAST 20TH AVE.
874-3641


DEBBIE JOYCE
1036 EAST 20TH AVE.
874-3641


DEBBIE JOYCE
1036 EAST 20TH AVE.
874-3641


DEBBIE JOYCE
1036 EAST 20TH AVE.
874-3641


DEBBIE JOYCE
1036 EAST 20TH AVE.
874-3641


DEBBIE JOYCE
1036 EAST 20TH AVE.
874-3641
```

```
]  SCRATCH L. SIMPSON BLK. D
NOV. 21/72 M

 2  CALL (11)
 5  LET I=9.000000E+37
 7  CALL (55,1)
 10  READ A,B,C
 15  IF A=0 THEN 9999
 17  CALL (9,1,1,R,S,T,U,V,W,X,Y,Z)
 20  IF A>R THEN 17
 25  IF A=R THEN 45
 30  PRINT "NO SUCH EMPLOYEE # AS"A
 33  PRINT
 40  GOTO 7
 45  LET P=INT((B*Z+C*Z*1.5-Y)*100+.5)/100
 50  FOR A=1 TO 12
 55  PRINT "*******";
 60  NEXT A
 65  PRINT
 70  PRINT TAB(7);"ACME ROAD-RUNNER ROAD GLUE CO,, LTD."
 75  PRINT TAB(42);"NOV. 21/72"
 80  PRINT
 85  PRINT TAB(14);"PAY $"P
 90  PRINT
 95  PRINT
 100  PRINT TAB(14);"TO "I;S;0;T;U;V;W;X;I
 105  PRINT
 110  PRINT
 115  PRINT TAB(6);"ACME BANK"
 120  PRINT TAB(6);"COYOTEVILLE NEVADA"
 125  PRINT
 130  FOR A=1 TO 12
 135  PRINT "*******";
 140  NEXT A
 150  PRINT
 155  PRINT
 160  GOTO 10
 165  DATA 1436,40,3
 170  DATA 2894,40,0
 175  DATA 3025,21,0
 180  DATA 3257,40,6
9998  DATA 0,0,0
9999  END

 RUN
**************************************************************************
       ACME ROAD-RUNNER ROAD GLUE CO,, LTD.
                                              NOV. 21/72

              PAY $ 102.05


              TO N VLAZY


       ACME BANK
       COYOTEVILLE NEVADA

**************************************************************************

NO SUCH EMPLOYEE # AS 3025
```

```
]  SCRATCH MR. OLSEN  KITS****
          =C

1  DATA 10,20,5,50
5  LET W=0
10  LET R=10
20  READ B
30  LET A=R/B
40  LET X=.5*A
50  LET Y=SQR(R↑2-X↑2)
60  LET D=A
65  IF (R↑2)-(X↑2)-(Y↑2) <= 0 THEN 90
70  LET Z=SQR(R↑2-X↑2-Y↑2)
80  LET T=Z*A*D
85  LET W=W+T
90  LET Y=Y-D
100  IF Y <= 0 THEN 120
110  GOTO 70
120  LET X=X+A
130  IF X >= R THEN 145
140  GOTO 50
145  LET W=W*B
150  PRINT "AREA=";W,"#STRIPS=";B
155  IF B#0 THEN 5
9999  END

 RUN D.L.J.
AREA= 4041.59                    #STRIPS= 10
AREA= 4139.37                    #STRIPS= 20
AREA= 3867.88                    #STRIPS= 5
AREA= 4181.1                     #STRIPS= 50
ERROR 56 IN LINE  20
```